

Activity Updates from the PHUSE Japan Open-Source Technology Team

PHUSE Japan OST at Japan SDE 2025



Our goal

- **Goal**

- ✓ To broaden the acceptance and general level of comfort with these technologies in our industry to assist in increasing their level of adoption in Japan.

- **Objective**

- ✓ Understand industry needs and HA requirements to Open-Source Technology (OST).
- ✓ Develop general guidance including implementation, use case and possible solution for clinical study / PMS for adoption of OST.
- ✓ Learn OST and suggest the way to implement OST in pharma industry.

Our team

PHUSE Japan OST Working Group

Masashi Aonuma (Astellas Pharma Inc.)	Ippei Akiya (ICON Clinical Research GK)
Naoko Izumi (Novartis Pharma K.K.)	Shunsuke Goto (Novartis Pharma K.K.)
Masashi Tanaka (EPS Corporation)	Yutaka Morioka (EPS Corporation)
Tomoyuki Namai (Chugai Pharmaceutical Co., Ltd.)	Tomoki Nishikawa (Kyowa Kirin Co., Ltd.)
Kensuke Sasaki (DATAACK Co., Ltd.)	Ryo Nakaya (Takeda Pharmaceutical Co., Ltd.)
Masataka Higashijima (Eli Lilly Japan K.K.)	Kazuyuki Miyashita (Nippon Boehringer Ingelheim Co., Ltd.)
Yasutaka Moriguchi (GlaxoSmithKline K.K.)	Yuichi Nakajima (GlaxoSmithKline K.K.)
Hiro Kosako (A2 Healthcare Corporation)	Hiroki Yamanobe (Maruho Co., Ltd.)



Our trajectory

- **SubTeam**
 - Use case consideration:
 - 1) Streamline our tasks,
 - 2) Actual case in PK analysis,
 - 3) Actual case in ARO.
- **OST Session at DIA Japan 2022**

- **SubTeam**
 - Learn and use Generate AI in clinical reporting.
- **Presentation at DIA Japan 2023**
- **R hands-on Training #1 Basic R**

- **SubTeam**
 - Pharmaforest – SAS package
 - R package development
 - R package validation
 - OMOP Standards
- **R hands-on Training #3 Package development**
- **Presentation at SAS User Group**

2020 **START**

- **Goal and Objective Setting.**
- **Team building.**

2022

8

2023

11

2025

16

2021

- **SubTeam**
 - Conduct OST survey.
 - Use case considerations: Testing GitHub / Sharing R sample programs
- **Presentation at PHUSE EU Connect 2021.**
- **OST Session at SAS User Group**

2024

14

- **SubTeam**
 - R submission
 - R package validation
 - Landscape Map of pharmaverse
 - Dataset-JSON
- **R hands-on Training #2 Rshiny**
- **Presentation at CDISC IC Japan, Regulatory Science Symposium.**

SAS Packages Development

PHUSE Japan OST

Ryo Nakaya (Takeda Pharmaceutical Company Limited)

Yutaka Morioka (EPS Corporation)

Hiroki Yamanobe (Maruho Co.,Ltd.)

05 Dec. 2025, PHUSE Japan SDE



Objectives

-Goal-

Through SAS Packages, we want to actively encourage sharing of SAS know-how that has often stayed within individuals. By doing this, we aim to build up collective knowledge, boost productivity, ensure quality through standardization, and energize our community.

-Priority-

Our first priority is to share openly—and get others to share as well—so that more people can join in. On that basis, we'll work on improving quality, driving standardization, and creating long-term value.



Objectives



Growth and Evolution of the SAS Programmer Community

**Fostering Collaboration Through Cross-Industry Sharing of SAS Expertise
in Drug Development**

PharmaForest Repository: Foundation and Key Initiatives

**SAS Code Packaging
with SAS Packages Framework**

**Real-Time Updates
Using LinkedIn and GitHub**

**The Community Is Also
Active in Japan**



PharmaForest : Packages

Over 30 SAS Packages
Available

A Wide Range of Packages:
Practical Utilities, Data & Log Validation,
and Creative Tools

One of the Packages Is Certified
by CDISC COSA.





PharmaForest Ecosystem Map

Output / Visualization



Other



Checker



Data Utility



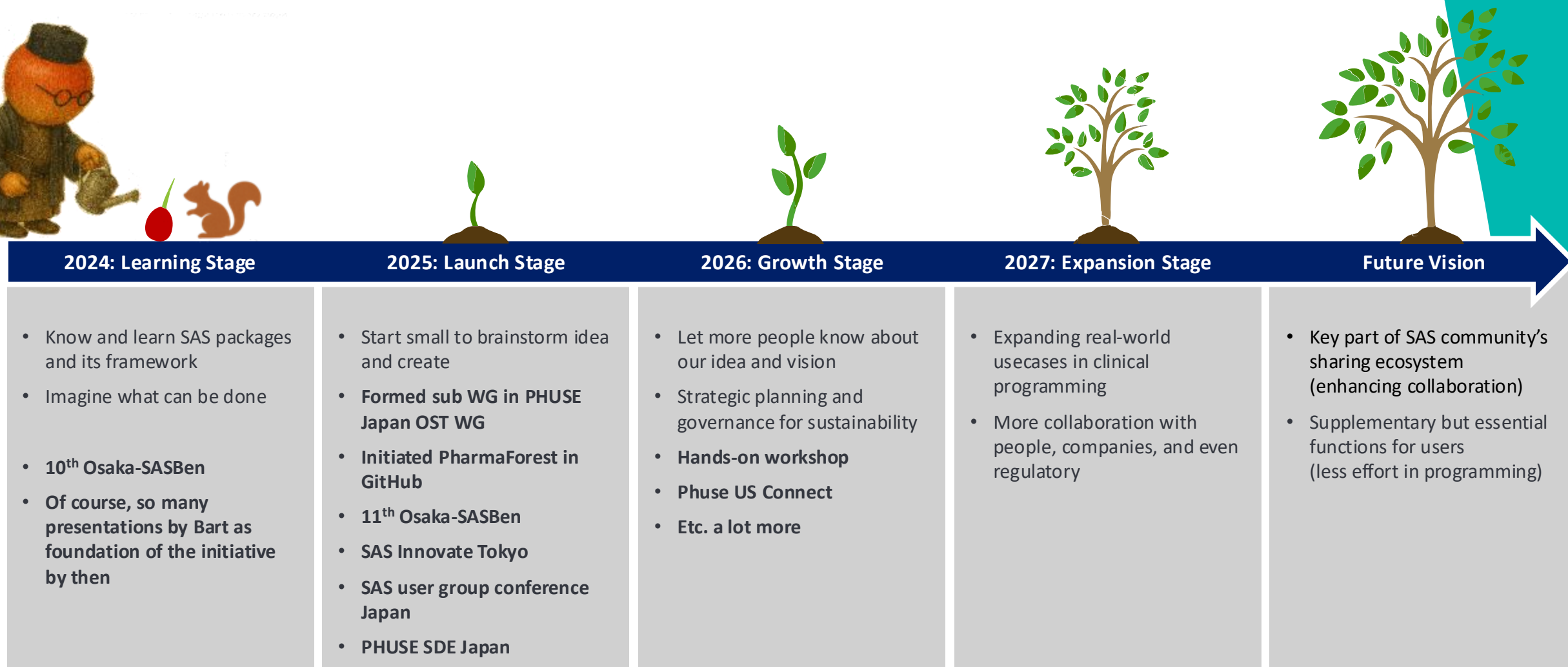
Forest Navigators

Provide Assistants to Help You Use
PharmaForest.
They Are AI Powered by MyGPTs of ChatGPT.

Two Types of Assistants:
1: Navigators for Package Guidance
2: Supporters for Package Creation



Our Journey



Open-source expansion and AI/ML evolution is transforming our way of programming in clinical development and lead us somewhere.
The initiative aims to lead the discussion by shaping ideas into forms, and envision/pursue future state.

Join Us! We Want You!



Use PharmaForest Packages



Let's Create Original Packages Together

Mirror Your Packages on PharmaForest

Join & Get the Latest Updates on LinkedIn

LinkedIn:
SAS Packages Users & Developers:
<https://www.linkedin.com/groups/10149811/>



Get Ready for Our Hands-On Workshop!

**SAS Package and PharmaFoest Hands-On Workshop
In 12th Osaka SAS Benkyo-Kai**

**2026-01-28 wed. 10:00~12:00 (JST)
Hybrid format: On-site and via Microsoft Teams**

第12回大阪SAS勉強会:

<https://sites.google.com/view/osakasasbenkyokai/>

Thank you



OMOP CDM

Slides Available Here!

English:

<https://nissinbo.github.io/phuse-sde-tokyo-2025-omop/en/>



Japanese:

<https://nissinbo.github.io/phuse-sde-tokyo-2025-omop/ja/>



Package development for PMDA Oncology inquiries

PHUSE OST

Shunsuke Goto

Hiro Kosako

Tomoyuki Namai

Hiroki Yamanobe

Yuichi Nakajima



Objectives

Deepen Understanding of Package Structure and Mechanism through Package Development as PHUSE -> Enhance Package Management

Create Industry-Wide Usable Packages and Promote Cross-Company Collaboration

Introduction

Challenges regarding to PMDA queries in oncology

- A large number of similar-format reports are required within a short period.
 - **No standardized format** like those defined by the FDA.
- > This leads to challenges in preparation and streamlining.

		Age group		
		例数 (%)		
PT (MedDRA ver.20.1)		2 歳以上 12 歳未満 29 例	12 歳以上 18 歳未満 38 例	18 歳以上 18 例
ALL AE	全有害事象*1			
	嘔吐	7 (24.1)	12 (31.6)	2 (11.1)
	食欲減退	9 (31.0)	6 (15.8)	2 (11.1)
	四肢痛	5 (17.2)	5 (13.2)	1 (5.6)
	疼痛	5 (17.2)	2 (5.3)	0
	医療機器関連感染	4 (13.8)	2 (5.3)	0
AE with Grade >=3	咽頭炎	2 (6.9)	4 (10.5)	0
	Grade 3 以上の有害事象*2			
	貧血	8 (27.6)	6 (15.8)	4 (22.2)
	医療機器関連感染	3 (10.3)	2 (5.3)	0
	好中球数減少	3 (10.3)	2 (5.3)	0
	白血球数減少	3 (10.3)	2 (5.3)	0
	血小板数減少	2 (6.9)	2 (5.3)	0
	疼痛	2 (6.9)	2 (5.3)	0
	AST 増加	2 (6.9)	0	0
	食欲減退	1 (3.4)	2 (5.3)	0
	胸水	0	2 (5.3)	0
	神経痛	0	2 (5.3)	0
	脱水	0	2 (5.3)	0
	発熱	0	2 (5.3)	0
	便秘	0	2 (5.3)	0
	重篤な有害事象*2			
	胸水	0	2 (5.3)	0
Serious AE	休業に至った有害事象*2			
	肺炎	2 (6.9)	0	0
	発熱	0	2 (5.3)	0

*1: 発現割合の差が 10%以上の事象を記載した、*2: 発現割合の差が 5%以上の事象を記載した

Overview of PMDA's Requests

#	Target studies	Request
1	All studies	Summary of AE overview • All AEs • AEs with grade 3+ • Fatal AEs • Serious AEs • AEs leading to treatment discontinuation • AEs leading to dose interruption/delay • AEs leading to dose reduction
2	All studies	Summary of AEs and drug-related AEs • The summaries for AEs and drug-related AEs need to be displayed on the one table. • The displays for following AEs are needed: Fatal AE, SAE, AE leading to treatment discontinuation of any treatment, AE leading to dose interruption/delay of any treatment, AE leading to dose reduction of any treatment
3	Studies including JPN	Summary of AE overview for Japanese and non-Japanese • AE overview for Japanese and non-Japanese needs to be displayed on the one table.
4	Pivotal studies	Summary of important AEs by time windows • The number and percentage of patients with each important AE needs to be summarized by time windows from study start.
...	...	...

Package to solve this challenge: {anaume}

- Analysis Utility for AE Mapping & EMBEDding
- Package to accelerate table generation for query from PMDA
- Using ARD to generate table



Analysis Results Data (ARD)

- Structured datasets that represent the outputs of statistical analyses.
- Designed in a reusable and standardized form.
 - ✓ Standardized column
 - ✓ Including Metadata
 - ✓ Machine-readable

{cards} data frame: 18 x 13

	group1	group1_level	variable	variable_level	context	stat_name	stat_label	stat	stat_fmt	fmt_fun	warning	error	gts_column
1	<NA>		ARM	A: Drug X categori...		n	n	134	134	0			stat_1
2	<NA>		ARM	A: Drug X categori...		N	N	400	400	0			stat_1
3	<NA>		ARM	A: Drug X categori...		p	% 0.335	33.5	<fn>				stat_1
4	<NA>		ARM	B: Place... categori...		n	n	134	134	0			stat_2
5	<NA>		ARM	B: Place... categori...		N	N	400	400	0			stat_2
6	<NA>		ARM	B: Place... categori...		p	% 0.335	33.5	<fn>				stat_2
7	<NA>		ARM	C: Combi... categori...		n	n	132	132	0			stat_3
8	<NA>		ARM	C: Combi... categori...		N	N	400	400	0			stat_3
9	<NA>		ARM	C: Combi... categori...		p	% 0.33	33.0	<fn>				stat_3
10	ARM	A: Drug X	AESER	SAE hierarch...		n	n	104	104	<fn>			stat_1
11	ARM	A: Drug X	AESER	SAE hierarch...		N	N	134	134	<fn>			stat_1
12	ARM	A: Drug X	AESER	SAE hierarch...		p	% 0.776	78	<fn>				stat_1
13	ARM	B: Place...	AESER	SAE hierarch...		n	n	101	101	<fn>			stat_2
14	ARM	B: Place...	AESER	SAE hierarch...		N	N	134	134	<fn>			stat_2
15	ARM	B: Place...	AESER	SAE hierarch...		p	% 0.754	75	<fn>				stat_2
16	ARM	C: Combi...	AESER	SAE hierarch...		n	n	99	99	<fn>			stat_3
17	ARM	C: Combi...	AESER	SAE hierarch...		N	N	132	132	<fn>			stat_3

Objectives of using ARD

Standardization

- Uniformity of summary table aggregation units, column names, and results between analyses

Efficiency

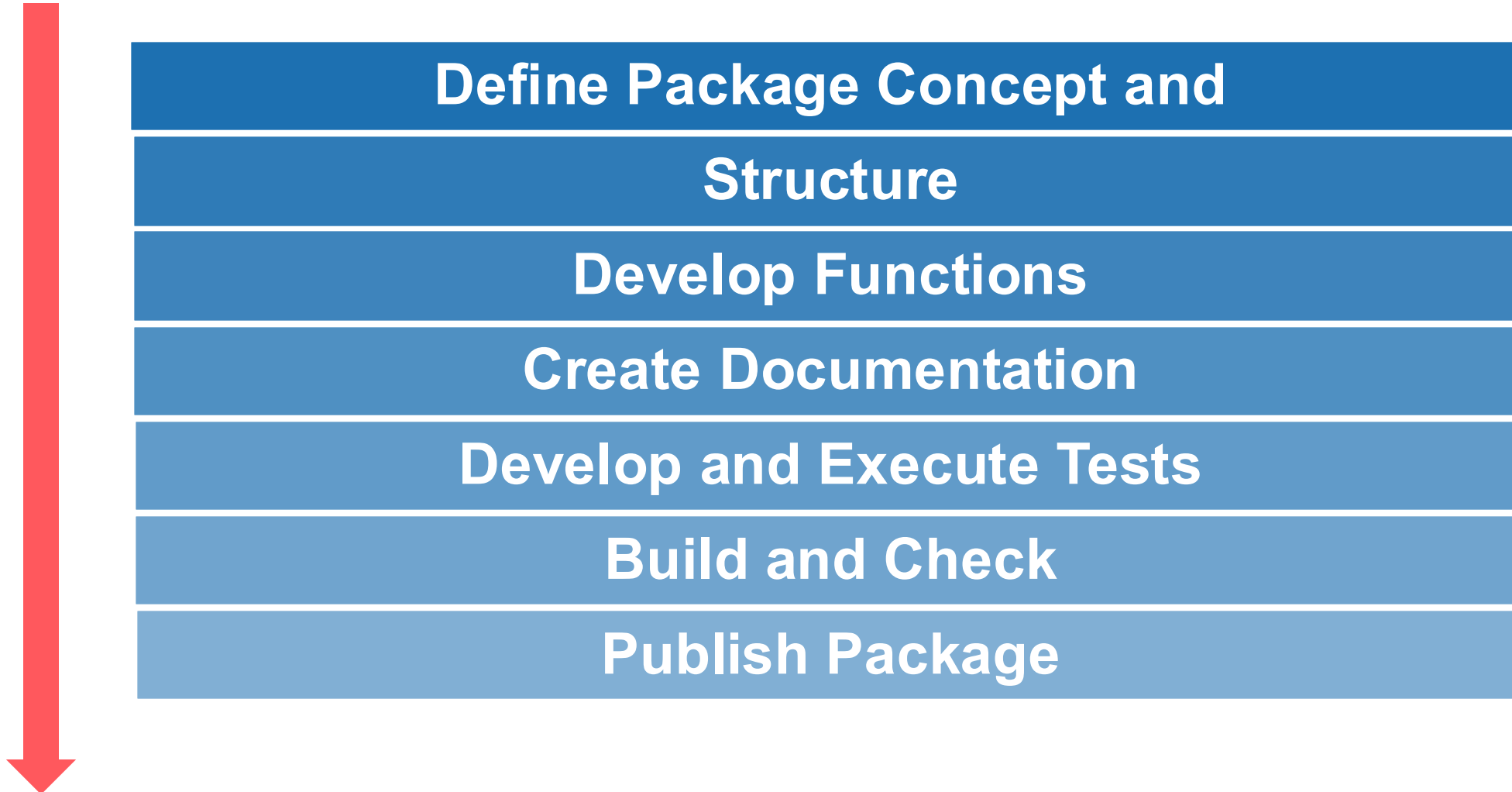
- Quick tabulation and re-formatting
- Enable QC easier

Reproductivity /Automation

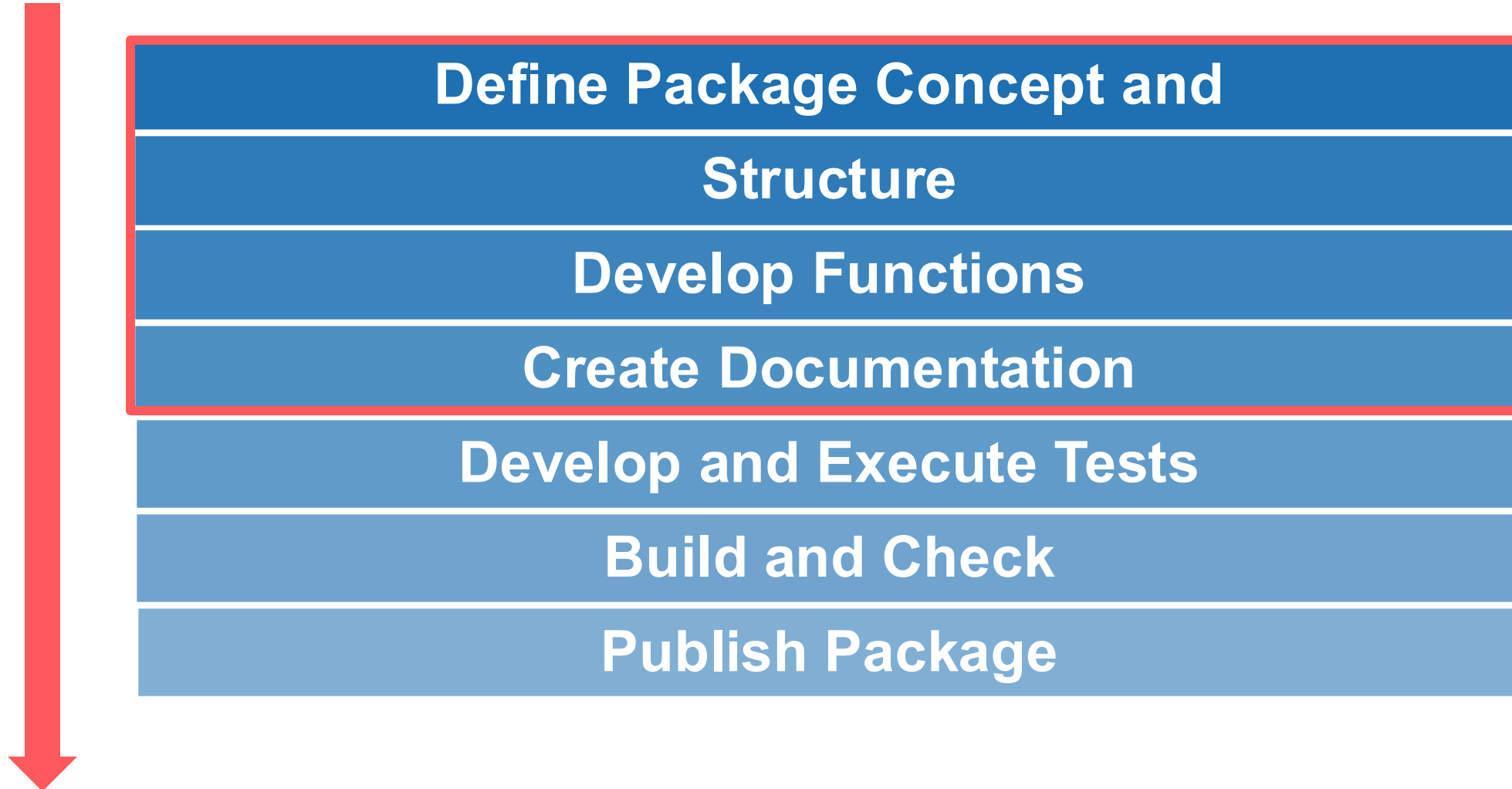
- The same analysis logic can be easily applied to other study or subgroup.
- ARD can be used in a variety of ways, such as in tables, R Shiny, figures, etc.

How to develop package

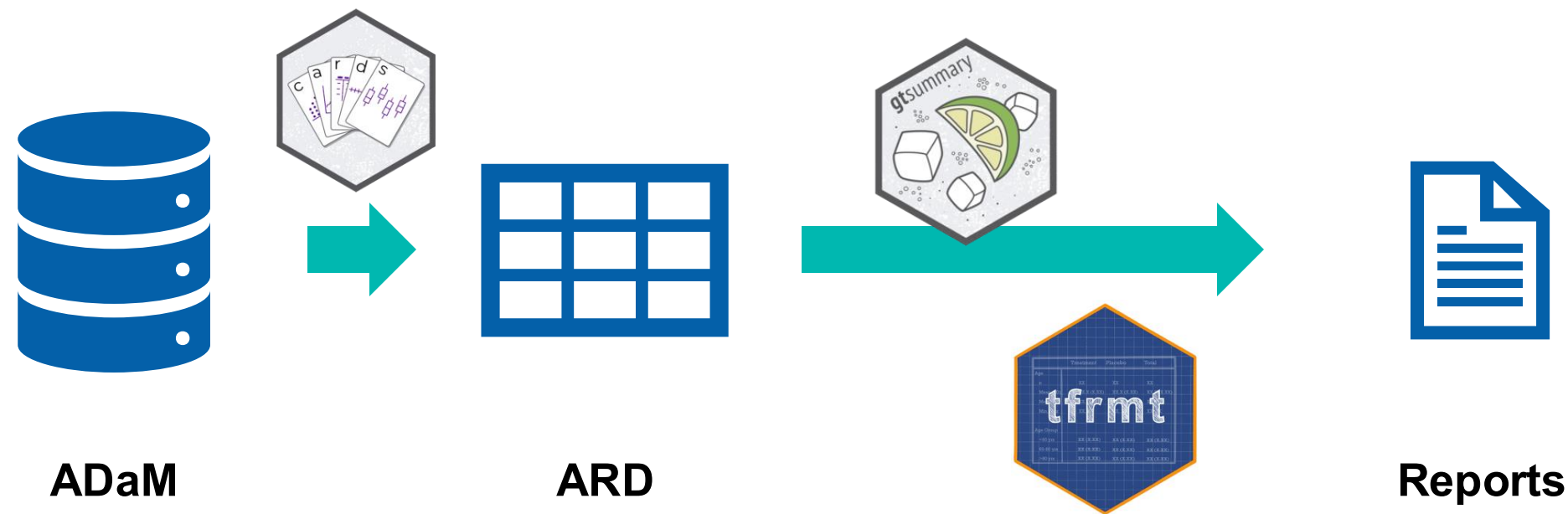
Package Development



Package Development



Working flow from data to reports using ARD



Using GitHub for development

The screenshot shows the GitHub interface for a repository named 'phuse_jp_test' owned by 'yuichi-nakaima'. The repository is private and has 3 watchers, 0 forks, and 0 stars. The main branch is 'main', with 8 branches and 0 tags. A search bar 'Go to file' is present. Below the repository header, a commit by 'stainlessfish' is shown, titled 'Update tfrmt_forfunc2.R', with commit hash '0f7eef3' and '15 Commits'.

File	Commit Message	Time
R	Update tfrmt_forfunc2.R	last month
data-raw	update code and dummy data	4 months ago
data	update code and dummy data	4 months ago
specification_files/libs	update code	3 months ago

On the right side, the 'About' section states 'No description, website, or topics provided.' and lists 'Activity', '0 stars', '3 watching', and '0 forks'. The 'Releases' section is also visible.

Overview of the package

{anaume}

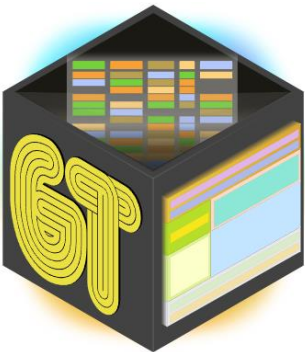
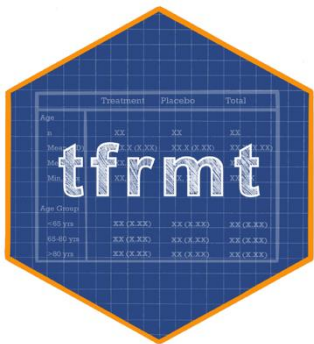
- Package to generate table
- functions
 - `make_ae_type()`
 - `jpn_query_overview()`
 - `jpn_query_summary_pt()`



Structure of ARDs for PMDA queries

- Sample ARD for “summary of AE overview”

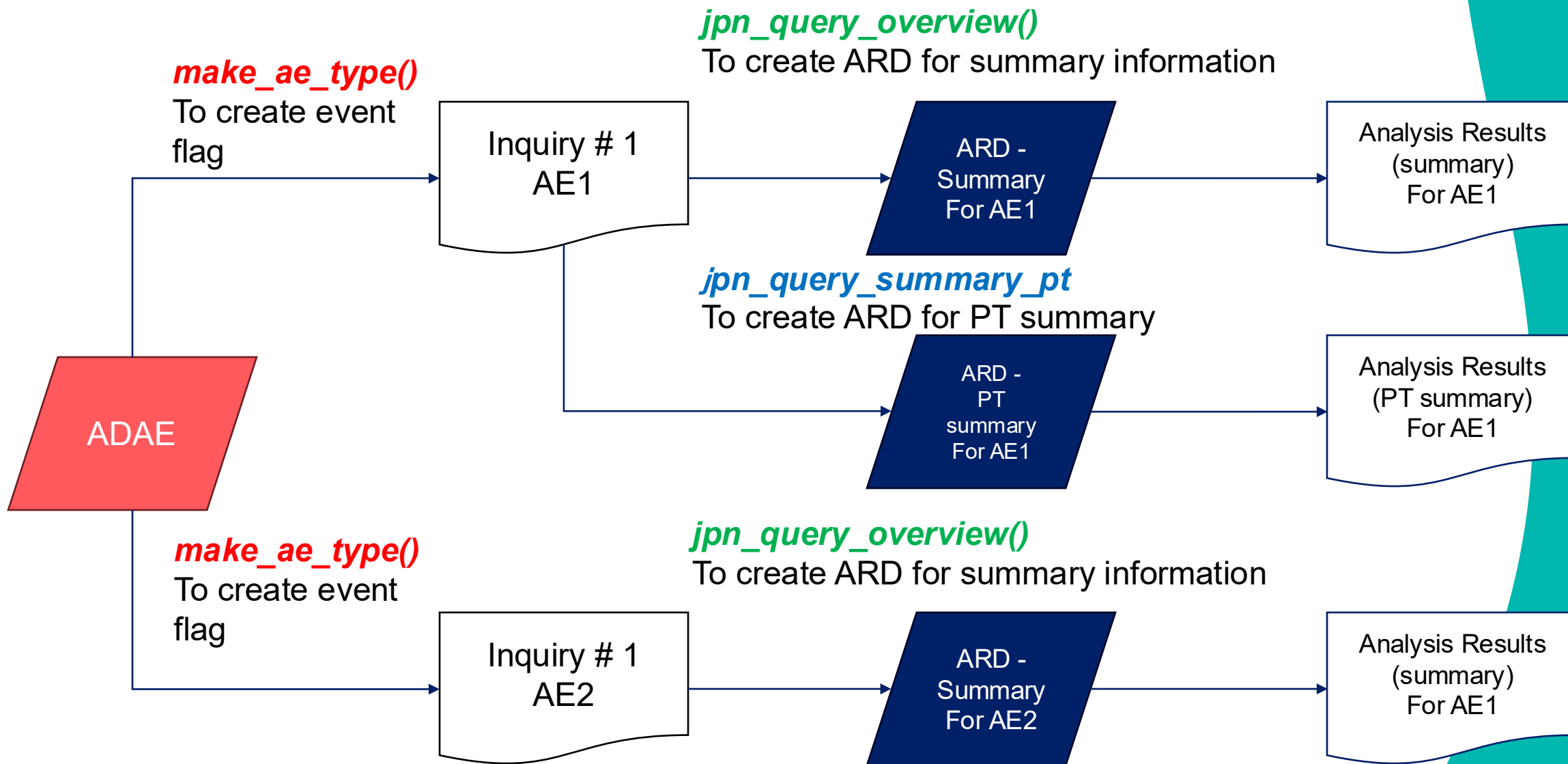
TRT01A	RACEGR1	Group	label	stat_name	stat_label	stat	ord1	ord2
Placebo	Non-white	Any AE	Any AE	n	n	6.000000	0	0
Placebo	Non-white	Any AE	Any AE	p	%	75.000000	0	0
Xanomeline High Dose	Non-white	Any AE	Any AE	n	n	9.000000	0	0
Xanomeline High Dose	Non-white	Any AE	Any AE	p	%	90.000000	0	0
Xanomeline Low Dose	Non-white	Any AE	Any AE	n	n	4.000000	0	0
Xanomeline Low Dose	Non-white	Any AE	Any AE	p	%	66.666667	0	0
Placebo	White	Any AE	Any AE	n	n	63.000000	0	0
Placebo	White	Any AE	Any AE	p	%	80.769231	0	0
Xanomeline High Dose	White	Any AE	Any AE	n	n	61.000000	0	0
1 Xanomeline High Dose	White	Any AE	Any AE	p	%	96.387097	0	0
1 Xanomeline Low Dose	White	Any AE	Any AE	n	n	82.000000	0	0
1 Xanomeline Low Dose	White	Any AE	Any AE	p	%	91.111111	0	0
1 Placebo	Non-white	Grade 3	AE >= Grade 3	n	n	6.000000	1	1
1 Placebo	Non-white	Grade 3	AE >= Grade 3	p	%	75.000000	1	1
1 Xanomeline High Dose	Non-white	Grade 3	AE >= Grade 3	n	n	9.000000	1	1
1 Xanomeline High Dose	Non-white	Grade 3	AE >= Grade 3	p	%	90.000000	1	1
1 Xanomeline Low Dose	Non-white	Grade 3	AE >= Grade 3	n	n	3.000000	1	1
1 Xanomeline Low Dose	Non-white	Grade 3	AE >= Grade 3	p	%	50.000000	1	1
1 Placebo	White	Grade 3	AE >= Grade 3	n	n	58.000000	1	1
2 Placebo	White	Grade 3	AE >= Grade 3	p	%	74.358974	1	1
2 Xanomeline High Dose	White	Grade 3	AE >= Grade 3	n	n	54.000000	1	1
2 Xanomeline High Dose	White	Grade 3	AE >= Grade 3	p	%	87.096774	1	1

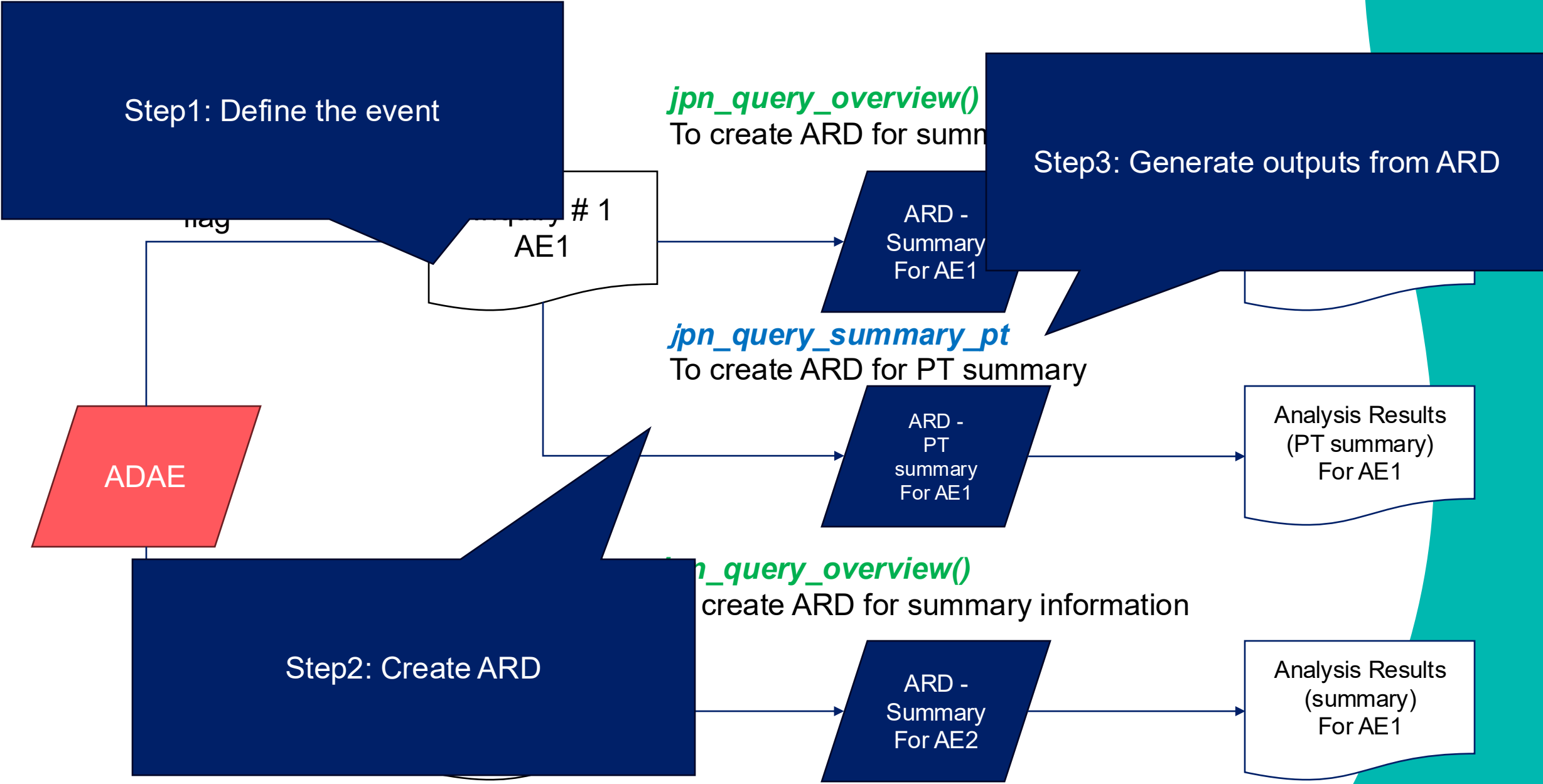


Adverse Event Summary

XXXX

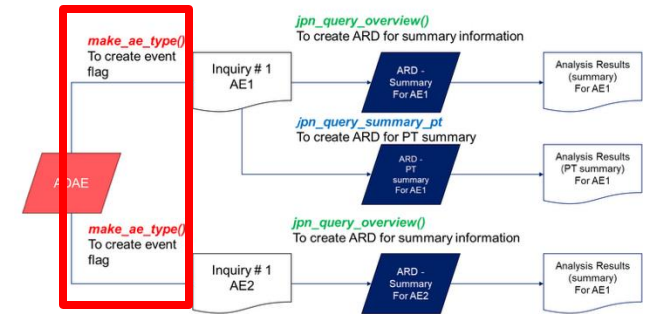
	Placebo (N=86)		Xanomeline High Dose (N=72)		Xanomeline Low Dose (N=96)	
	Non-white	White	Non-white	White	Non-white	White
Any AE	6 (75%)	63 (80.8%)	9 (90%)	61 (98.4%)	4 (66.7%)	82 (91.1%)
AE >= Grade 3	6 (75%)	58 (74.4%)	9 (90%)	54 (87.1%)	3 (50%)	80 (88.9%)
Any SAE	NA (NA)	NA (NA)	NA (NA)	1 (1.6%)	NA (NA)	2 (2.2%)
AE leads drug withdraw	3 (37.5%)	41 (52.6%)	6 (60%)	38 (61.3%)	2 (33.3%)	60 (66.7%)
AE leads drug interrupt	6 (75%)	36 (46.2%)	8 (80%)	45 (72.6%)	1 (16.7%)	62 (68.9%)
AE leads to dose reduction	4 (50%)	35 (44.9%)	8 (80%)	48 (77.4%)	3 (50%)	58 (64.4%)





Step1: Define the event make_ae_type

- Define conditions for target AEs and display labels for tables
- Create an object of class ae_type (essentially a list)
- Pass this object as an argument to subsequent functions for flexible analysis



```
> make_ae_type("aeg3", "AE >= Grade 3", expr(AEGRD == "Y"))
```

ae_type

id: aeg3

label: AE >= Grade 3

condition: AEGRD == "Y"

Label to be displayed

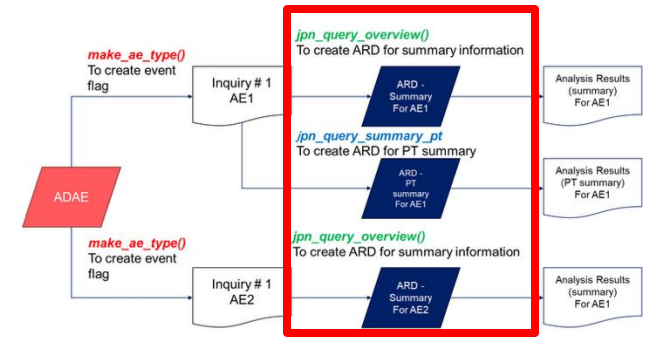
Condition for this ae type

```
> ae_types <- list(
+   make_ae_type("any_ae", "Any AE", expr(TRUE)),
+   make_ae_type("aeg3", "AE >= Grade 3", expr(AEGRD == "Y")),
+   make_ae_type("sae", "Any SAE", expr(AESER == "Y")),
+   make_ae_type("fatal", "Fatal SAEs", expr(AESER == "Y" & AEFAT == "Y")),
+   make_ae_type("disc", "AE leads drug withdraw", expr(AEDISCON == "Y")),
+   make_ae_type("itrr", "AE leads drug interrupt", expr(AEITRR == "Y")),
+   make_ae_type("redu", "AE leads to dose reduction", expr(AEREDUCE == "Y"))
+ )
```

Step2: Create ARD

jpn_query_overview/jpn_query_summary_pt

- `jpn_query_overview()`
 - Function to Generate ARD for AE Overview



- `jpn_query_summary_pt()`
 - Function to Generate ARD for AE by PT

Both functions internally use {cards}.

表 11 安全性の概要 (ALINA 試験、2023 年 6 月 26 日データカットオフ)

	例数 (%)	
	本薬群 128 例	Chemo 群 120 例
全有害事象	126 (98.4)	112 (93.3)
Grade 3 以上の有害事象	38 (29.7)	37 (30.8)
死亡に至った有害事象	0	0
重篤な有害事象	17 (13.3)	10 (8.3)
投与中止に至った有害事象	7 (5.5)	15 (12.5)
休薬に至った有害事象	35 (27.3)	22 (18.3)
減量に至った有害事象	33 (25.8)	12 (10.0)

表 10 18 歳以上の集団と比較して、2 歳以上 12 歳未満又は 12 歳以上 18 歳未満の集団で発現割合が高かった有害事象 (GO29664 試験、安全性解析対象、2018 年 4 月 2 日データカットオフ)

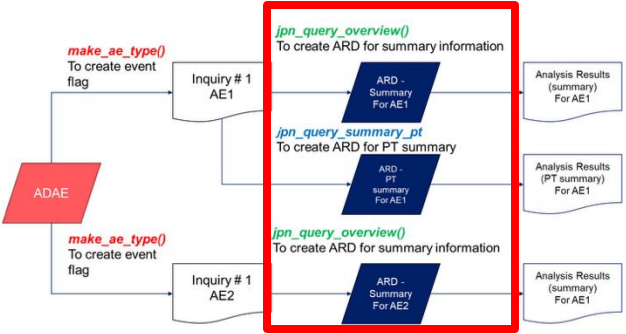
PT (MedDRA ver.20.1)	例数 (%)		
	2 歳以上 12 歳未満 29 例	12 歳以上 18 歳未満 38 例	18 歳以上 18 例
全有害事象 ¹			
嘔吐	7 (24.1)	12 (31.6)	2 (11.1)
食欲減退	9 (31.0)	6 (15.8)	2 (11.1)
四肢痛	5 (17.2)	5 (13.2)	1 (5.6)
疼痛	5 (17.2)	2 (5.3)	0
医療機器関連感染	4 (13.8)	2 (5.3)	0
咽頭炎	2 (6.9)	4 (10.5)	0
Grade 3 以上の有害事象 ²			
貧血	8 (27.6)	6 (15.8)	4 (22.2)
医療機器関連感染	3 (10.3)	2 (5.3)	0
好中球数減少	3 (10.3)	2 (5.3)	0
白血球数減少	3 (10.3)	2 (5.3)	0
血小板数減少	2 (6.9)	2 (5.3)	0
疼痛	2 (6.9)	2 (5.3)	0
AST 増加	2 (6.9)	0	0
食欲減退	1 (3.4)	2 (5.3)	0

Step2: Create ARD

ex) jpn_query_overview

- jpn_query_overview()

```
> jpn_query_overview(  
+   data = adae,  
+   adsl = adsl,  
+   ae_types = ae_types,  
+   strat_var = NULL #stratification_var  
+ )
```



	TRT01A	group	label	stat_name	stat_label	stat	ord1	ord2
1	Placebo	Any AE	Any AE	n	n	69.000000	0	0
2	Placebo	Any AE	Any AE	p	%	80.232558	0	0
3	Xanomeline High Dose	Any AE	Any AE	n	n	70.000000	0	0
4	Xanomeline High Dose	Any AE	Any AE	p	%	97.222222	0	0
5	Xanomeline Low Dose	Any AE	Any AE	n	n	86.000000	0	0
6	Xanomeline Low Dose	Any AE	Any AE	p	%	89.583333	0	0
7	Placebo	Grade 3	AE >= Grade 3	n	n	64.000000	1	1
8	Placebo	Grade 3	AE >= Grade 3	p	%	74.418605	1	1
9	Xanomeline High Dose	Grade 3	AE >= Grade 3	n	n	63.000000	1	1
10	Xanomeline High Dose	Grade 3	AE >= Grade 3	p	%	87.500000	1	1
11	Xanomeline Low Dose	Grade 3	AE >= Grade 3	n	n	83.000000	1	1
12	Xanomeline Low Dose	Grade 3	AE >= Grade 3	p	%	86.458333	1	1
13	Xanomeline High Dose	Any SAE	Any SAE	n	n	1.000000	1	0
14	Xanomeline High Dose	Any SAE	Any SAE	p	%	1.388889	1	0
15	Xanomeline Low Dose	Any SAE	Any SAE	n	n	2.000000	1	0
16	Xanomeline Low Dose	Any SAE	Any SAE	p	%	2.083333	1	0

{cards} package

- Part of Pharmaverse
- R package to create ARD



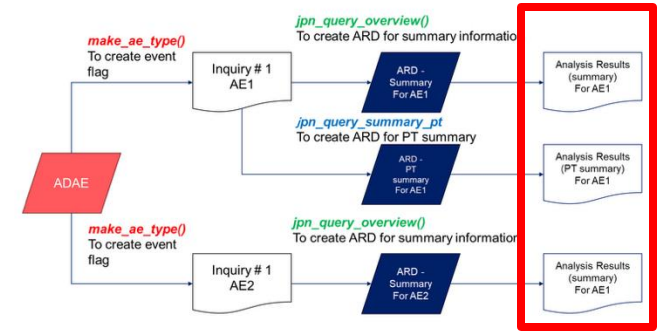
```
ard_categorical(  
  data = adae,  
  by = c('AGEGR1','TRT01A'),  
  variables = AEDECOD,  
  denominator = adsl  
)
```

	group1	group1_level	group2	group2_level	variable	variable_level	stat_name	stat_label	stat
1	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	n	n	0
2	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	N	N	72
3	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	p	%	0
4	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	n	n	1
5	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	N	N	72
6	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ABDOMINA...	p	%	0.014
7	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ACROCHOR...	n	n	0
8	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ACROCHOR...	N	N	72
9	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ACROCHOR...	p	%	0
10	AGEGR1	>64	TRT01A	Placebo	AEDECOD	ACTINIC ...	n	n	0

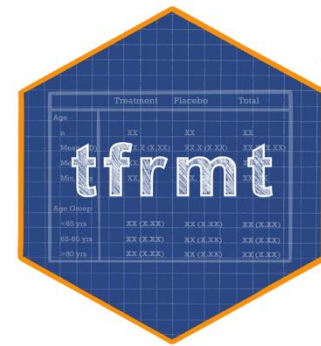
i 4346 more rows

Step3: Create outputs from ARD

- Provide template program to generate table from ARD instead of functions for flexible modification in actual queries.
- Generate Output from ARD using:
 - ✓ {gt}
 - ✓ {gtsummary}
 - ✓ {tfrmt}
- Next Slides Will Show Examples Using {tfrmt}



How to generate outputs from ARD : {tfrmt}

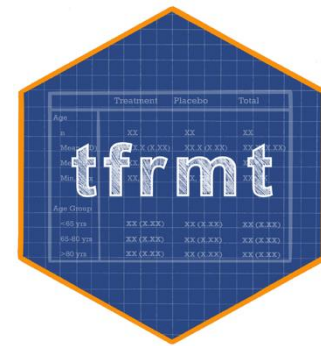


```
library(dplyr, tfrmt, gt)
# setting tfrmt
tf <- tfrmt(
  column      = c(TRT01A),
  group       = group,
  label       = label,
  param       = stat_name,
  value       = stat,
  sorting_cols = c(ord1, ord2),
  big_n       = bn,
  body_plan   = bp,
  # col_plan   = cp,
  col_plan    = col_plan(
    span_structure(
      TRT01A = c("Xanomeline Low Dose", "Xanomeline High
Dose", "Placebo")
    )
  )
  # col_style_plan = csp
)

# print to gt
gt_tbl <- print_to_gt(tf, tbl_disp) %>%
  cols_hide(columns = c(ord1, ord2, stat_label)) %>%
  tab_header(
    title = "PMDA inquiries",
    subtitle = "Adverse Event"
  )
```

PMDA inquiries			
Adverse Event			
	Placebo N = 86	Xanomeline High Dose N = 72	Xanomeline Low Dose N = 96
Any AE	69 (80.2%)	70 (97.2%)	86 (89.6%)
Any SAE		1 (1.4%)	2 (2.1%)
Grade 3			
AE >= Grade 3	64 (74.4%)	63 (87.5%)	83 (86.5%)
Dose related			
AE leads drug withdraw	44 (51.2%)	44 (61.1%)	62 (64.6%)
AE leads drug interrupt	42 (48.8%)	53 (73.6%)	63 (65.6%)
AE leads to dose reduction	39 (45.3%)	56 (77.8%)	61 (63.5%)

How to generate outputs from ARD : {tfrmt}



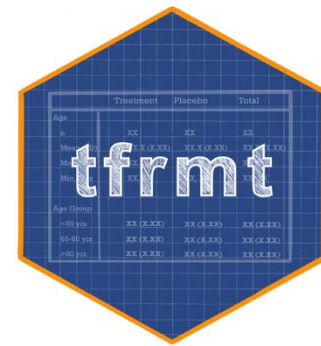
```
library(dplyr, tfrmt, gt)
# setting tfrmt
tf <- tfrmt(
  column      = c(TRT01A),
  group       = group,
  label       = label,
  param       = stat_name,
  value       = stat,
  sorting_cols = c(ord1, ord2),
  big_n       = bn,
  body_plan   = bp,
  # col_plan   = cp,
  col_plan    = col_plan(
    span_structure(
      TRT01A = c("Xanomeline Low Dose", "Xanomeline High
Dose", "Placebo")
    )
  )
  # col_style_plan = csp
)

# print to gt
gt_tbl <- print_to_gt(tf, tbl_disp) %>%
  cols_hide(columns = c(ord1, ord2, stat_label)) %>%
  tab_header(
    title = "PMDA inquiries",
    subtitle = "Adverse Event"
  )
```

PMDA inquiries			
Adverse Event			
	Placebo N = 86	Xanomeline High Dose N = 72	Xanomeline Low Dose N = 96
Any AE	69 (80.2%)	70 (97.2%)	86 (89.6%)
Any SAE		1 (1.4%)	2 (2.1%)
Grade 3			
AE >= Grade 3	64 (74.4%)	63 (87.5%)	83 (86.5%)
Dose related			
AE leads drug withdraw	44 (51.2%)	44 (61.1%)	62 (64.6%)
AE leads drug interrupt	42 (48.8%)	53 (73.6%)	63 (65.6%)
AE leads to dose reduction	39 (45.3%)	56 (77.8%)	61 (63.5%)

Step 1: Define the Table Format

How to generate outputs from ARD : {tfrmt}



```
library(dplyr, tfrmt, gt)
# setting tfrmt
tf <- tfrmt(
  column      = c(TRT01A),
  group       = group,
  label       = label,
  param       = stat_name,
  value       = stat,
  sorting_cols = c(ord1, ord2),
  big_n       = bn,
  body_plan   = bp,
  # col_plan   = cp,
  col_plan    = col_plan(
    span_structure(
      TRT01A = c("Xanomeline Low Dose", "Xanomeline High
Dose", "Placebo")
    )
  )
  # col_style_plan = csp
)

# print to gt
gt_tbl <- print_to_gt(tf, tbl_disp) %>%
  cols_hide(columns = c(ord1, ord2, stat_label)) %>%
  tab_header(
    title = "PMDA inquiries",
    subtitle = "Adverse Event"
  )
```

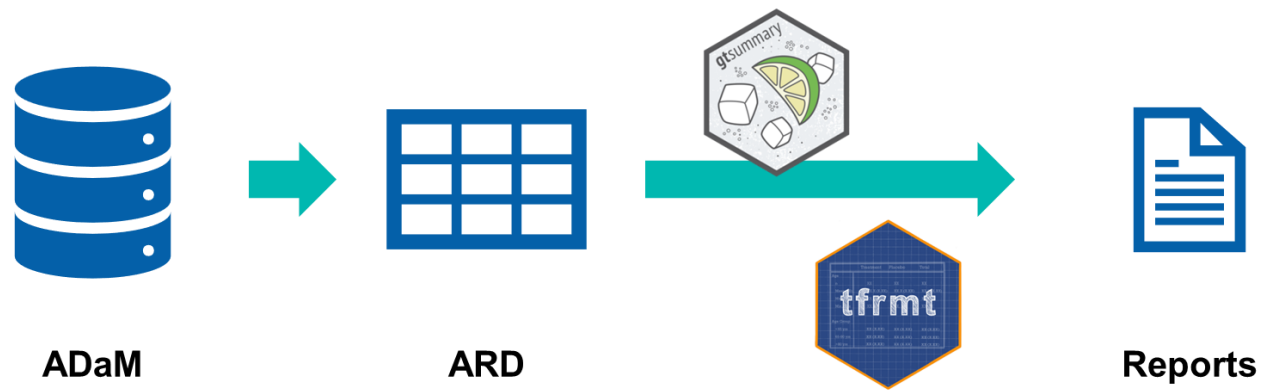
	PMDA inquiries		
	Adverse Event		
	Placebo N = 86	Xanomeline High Dose N = 72	Xanomeline Low Dose N = 96
Any AE	69 (80.2%)	70 (97.2%)	86 (89.6%)
Any SAE		1 (1.4%)	2 (2.1%)
Grade 3			
AE >= Grade 3	64 (74.4%)	63 (87.5%)	83 (86.5%)
Dose related			
AE leads drug withdraw	44 (51.2%)	44 (61.1%)	62 (64.6%)
AE leads drug interrupt	42 (48.8%)	53 (73.6%)	63 (65.6%)
AE leads to dose reduction	39 (45.3%)	56 (77.8%)	61 (63.5%)

Step 2: Apply the Format to Data and Convert to gt Object

Closing

Summary

- By using ARD, flexible tables can be easily generated.
- We propose a method using ARDs to efficiently conduct extensive and non-formatted analyses required by PMDA.



Future outlook

- Updating template programs to generate outputs from ARD
- Creating document vignettes and tests
- [stretching] Release this package
- [stretching] Utilize ARD for R Shiny app

Risk Assessment

Hiro Kosako
on behalf of Phuse Japan OST WG
(Masashi Aonuma, Masashi Tanaka,
Shunsuke Goto, Tomoki Nishikawa,
Hiro Kosako, Ryo Nakaya)



Phuse Japan Open-Source Technology WG (Package Risk Assessment Sub-team)

- Masashi Aonuma (Astellas Pharmaceutical)
- Masashi Tanaka (EPS Corporation)
- Shunsuke Goto (Novartis Pharma K.K.)
- Tomoki Nishikawa (Kyowa Kirin)
- Ryo Nakaya (Takeda Pharmaceutical)
- Hiro Kosako (A2 Healthcare)

Disclaimer:

The views and opinions expressed in this presentation are those of the authors and do not necessarily reflect the official policy or position of any affiliated organizations or employers.



Executive Summary

It is important to set-up validation strategy of analysis systems in drug development in pharmaceutical industry. One of the hot topics in open-source software(OSS) utilization in this space is **R package validation**.

Since risk-based approach of the validation is commonly employed, we **summarize the available tools (e.g. riskmetric, etc.) for risk assessment** in the approach and discuss the points to consider in the presentation.



Agenda

- Background
- Risk-based R package validation
- Tools
- Key takeaways



Background



Proprietary Software

Procedures / functions
validated by SAS inc.



Open-Source Software

Packages from various
contributors

Types of packages

Base packages
(base, utils, ...)

Included with R installation
(installed and loaded)

Bundled with R installation
(installed but not loaded by default)

Recommended packages
(matrix, survival, ...)

It is considered Minimal Risk based on the following points:

- The R Foundation ensures accuracy through the Software Development Life Cycle (SDLC).
- Bugs are fixed in response to feedback from the R community.

Contributed packages
(ggplot2, dplyr, ...)

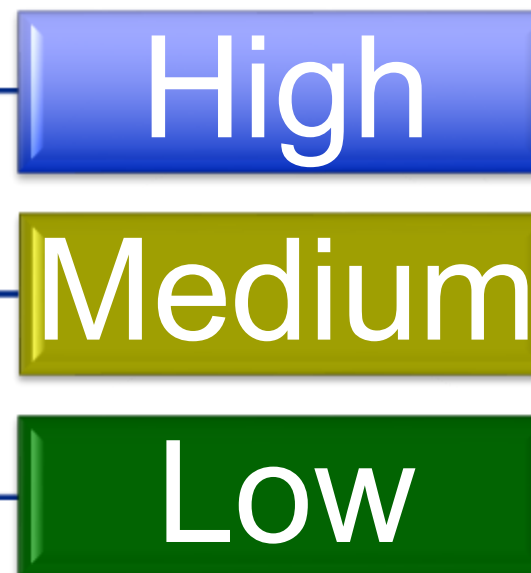
Need to install from CRAN, Github,
or Bioconductor

It is considered necessary to conduct a risk evaluation.

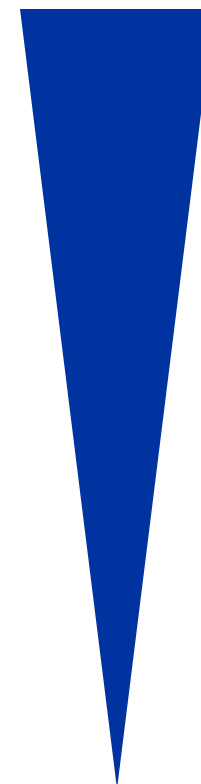
Risk-based R package validation



Risk Assess



Resource for
validation



As of 21October2025, CRAN has **22,931** contributed packages



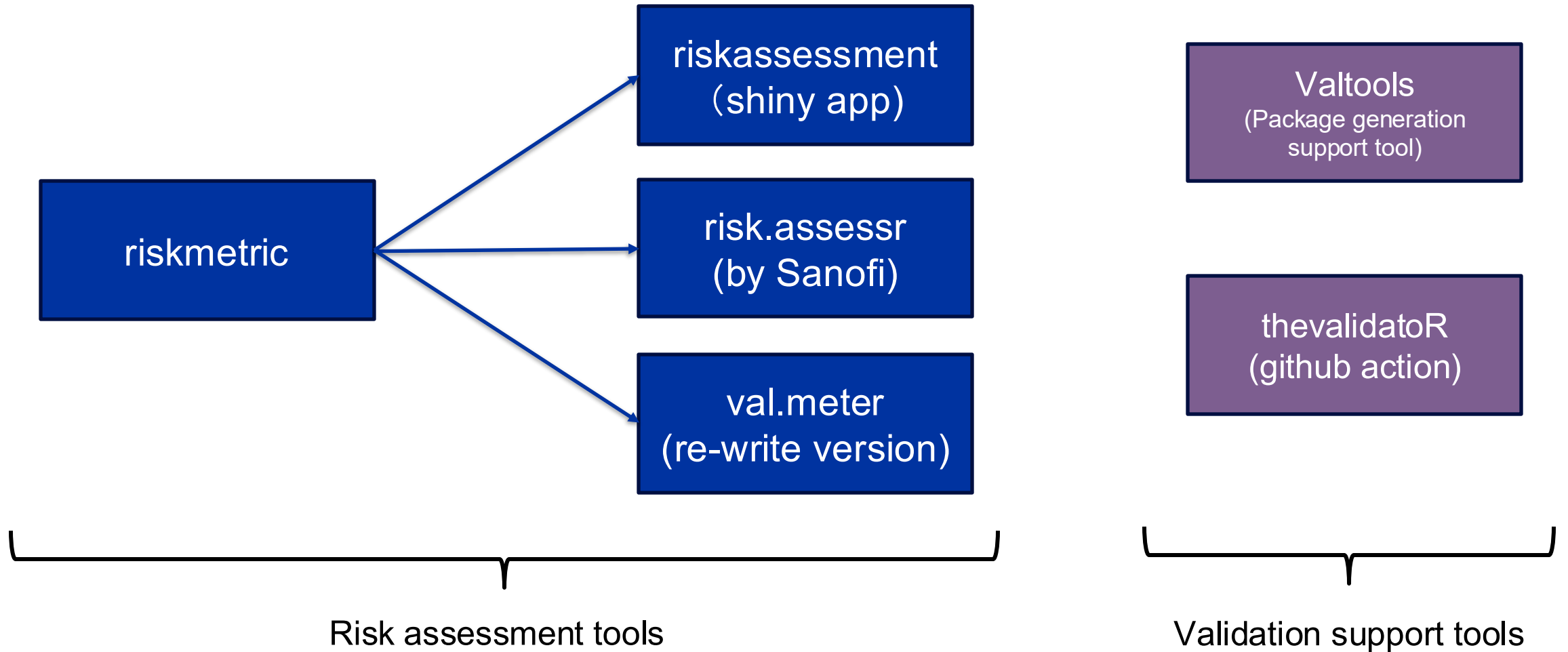
Tools

- riskmetric
- riskassessment
- val.meter
- risk.assessr
- valtools
- thevalidatorR

Note: These are not exhaustive. More tools exist(e.g. mpn.scorecard).



Tools





riskmetric

R package
{riskmetric}





Overview

- A collection of risk metrics to evaluate the quality of R packages
- *This package is in experimentation. Final considerations about design are being considered, but core concepts are considered final.*

Reference

<https://pharmar.github.io/riskmetric/>

<https://cran.r-project.org/web/packages/riskmetric/refman/riskmetric.html>

Example

```
pkg_ref(c("riskmetric", "utils", "tools")) |>  
pkg_assess() |>  
pkg_score() # Scores assessment criteria.
```

Tips

`pkg_assess()` is useful to get package information, e.g., URLs.



Metrics of riskmetric (1/2)

Metric	Description
covr_coverage	Code coverage of the entire target package obtained via <code>covr::coverage_to_list()</code> . <i>The return values unexpectedly became NA*.</i>
has_news	Whether the package contains a NEWS file.
remote_checks	Score calculated from the number of OS flavors that passed with OK or NOTES and the number of OS's that produced WARNINGS
news_current	Whether the current version number is mentioned in the NEWS file
r_cmd_check	Score calculate from the number of notes, errors, and warnings. The score is the weighted sum of notes (0.1), errors (1) and warnings (0.25). <i>The return values unexpectedly became NA*.</i>
exported_namespace	Score calculated from the number of exported objects [0,1]
has_vignettes	Whether the package contains vignettes
export_help	Proportion of exported objects in the NAMESPACE that have documentation.

* The metrics became NA when using `pkg_assess()` to all packages in {pharmaverse} as of 3Oct2025.



Metrics of riskmetric (2/2)

Metric	Description
has_website	1 if any website url is provided, otherwise 0 E.g., admiral: https://pharmaverse.github.io/admiral/ , https://github.com/pharmaverse/admiral
has_maintainer	1 if any maintainer is provided, otherwise 0
bugs_status	A fractional value indicating percentage of last 30 bug reports that are now closed <i>When performing many repeated assessments, the return values unexpectedly became NA as of 3Oct2025.</i>
size_codebase	Numeric value between 0 (for large codebase) and 1 (for small codebase)
has_source_control	Returns 1 if any source control url is provided, otherwise 0 E.g., admiral: https://github.com/pharmaverse/admiral
has_bug_reports_url	A logical value indicating whether the package has a BugReports field filled in
downloads_1yr	Numeric value between 0 (low) and 1 (high download volume) converting the number of downloads E.g., admiral: $1-150000/(31338+150000)=0.1728154$
reverse_dependencies	Numeric value between 1 (high number of reverse dependencies) and 0 (low number of reverse dependencies) The possible minimum value is $0.2463761=1/(1+\exp(-0.5*\sqrt{5}))$. E.g., admiral: $1/(1+\exp(-0.5*(\sqrt{7}-\sqrt{5})))=0.5510321$
has_examples	1 if any example or usage fields are found, otherwise 0
dependencies	Numeric value between 0 (high number of dependencies) and 1 (low number of dependencies) The possible maximum value is $0.8807971=1-1/(1+\exp(-0.5*4))$. E.g., admiral: $1-1/(1+\exp(-0.5*(12-4)))=0.01798621$
license	Score of metric license <i>The return values unexpectedly became NA*.</i>



How to Calculate pkg_score

pkg_score for {admiral} = 0.49794082

Metric	Original	Replacing NA with zero
covr_coverage	NA	0
has_news	1	
remote_checks	1	
news_current	1	
r_cmd_check	NA	0
exported_namespace	NA	0
has_vignettes	1	
export_help	NA	0
has_website	1	
has_maintainer	1	
bugs_status	0.8	
size_codebase	NA	0
has_source_control	1	
has_bug_reports_url	1	
downloads_1yr	0.170106116	
reverse_dependencies	0.551032099	
has_examples	NA	0
dependencies	0.01798621	
license	NA	0

Average
0.50205918

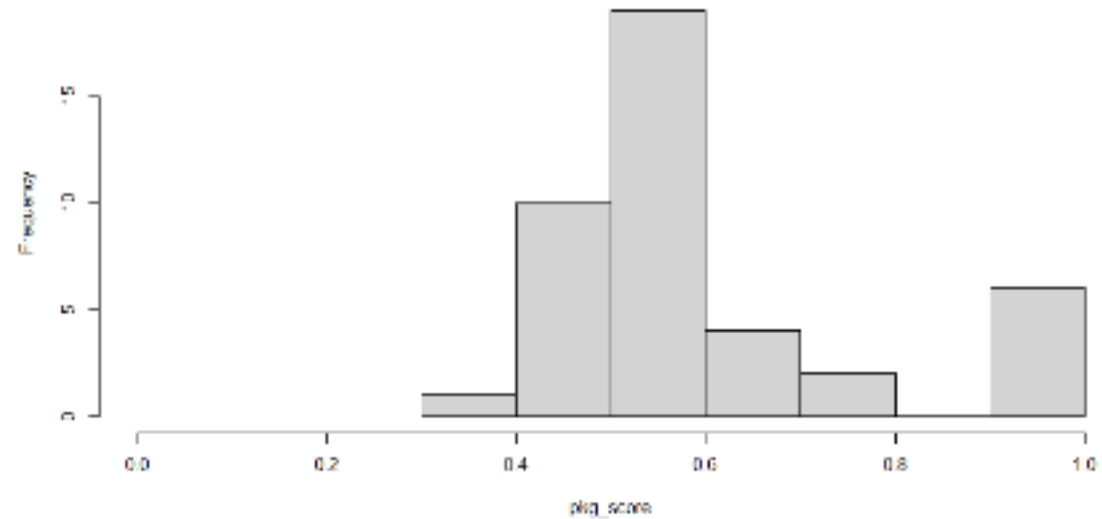
1 minus Average
0.49794082

	0	1
Each metric	Not very good	Good
pkg_score	Good	Not very good

One metric affects approximately 0.05 (=1/19) on pkg_score.
When comparing two packages, the delta pkg_score around 0.1 may not be critical.

Category	Package	Version	pkg_score (min-max)
ADaM	admiral	1.3.1	0.49–0.50
esub	pkglite	0.2.2	0.39–0.39
esub	xportr	0.4.3	0.53–0.53
metadata	metacore	0.1.2	0.43–0.43
metadata	metatools	0.1.5	0.57–0.57
python	rtflite	NA	0.99–0.99
sdtm	aNCA	NA	0.99–0.99
sdtm	datacutr	0.2.3	0.52–0.52
sdtm	sdtm.oak	0.2.0	0.53–0.53
sdtm	sdtmchecks	1.0.0	0.54–0.54
synthetic	pharmaverseadam	0.2.0	0.51–0.52
synthetic	pharmaverseraw	0.1.1*	0.62–0.62
synthetic	pharmaversesdtm	1.0.0	0.51–0.51
tlg	autoslider	NA	0.99–0.99
tlg	cards	0.7.0*	0.51–0.52
tlg	cardx	0.3.0*	0.53–0.53
tlg	chevron	0.2.12	0.52–0.52
tlg	clinify	0.3.0	0.77–0.77
tlg	ggsurvfit	1.2.0*	0.48–0.49
tlg	gtsummary	2.4.0*	0.45–0.46
tlg	pharmaRTF	0.1.4	0.61–0.61
tlg	rhino	1.10.1	0.46–0.46
tlg	rlistings	0.2.12	0.48–0.49
tlg	rtables	0.6.9	0.45–0.46
tlg	teal	1.0.0*	0.51–0.52
tlg	tern	0.9.5	0.45–0.45
tlg	tfrmt	0.2.0*	0.59–0.62
tlg	tfrmtbuilder	0.1.0	0.58–0.58
tlg	tidyCDISC	0.2.1	0.54–0.54
tlg	tidytlg	0.10.0	0.51–0.51
tlg	Tplyr	1.2.1	0.53–0.56
utilities	connector	1.0.0	0.51–0.58
utilities	datasetjson	0.3.0	0.54–0.58
utilities	envsetup	NA	0.99–0.99
utilities	logrx	0.3.1	0.41–0.45
utilities	whirl	0.3.1*	0.52–0.58
validation	covtracer	0.0.1	0.49–0.53
validation	risk.assessr	2.0.1*	0.62–0.62
validation	riskassessment	NA	0.99–0.99
validation	riskmetric	0.2.4	0.49–0.49
validation	valtools	NA	0.99–0.99
-	base	4.3.2	0.75–0.75

* Versioned up between August 20 and October 3, 2025.



Assessed pharmaverse packages and {base} on Aug. 20, Sept. 11, and Oct. 3, 2025.

The pkg_score shows medium term stability; the widest range is 0.51–0.58 for {connector}.



riskassessment

R package
{riskassessment}



Overview

{riskassessment} package / shinyapp. by R Validation Hub



- Analyze {riskmetric} outputs without the need to write code.
- Categorize a package with an overall assessment (i.e., low, medium, or high risk).
- Add personalized comments on individual metrics.
- Download reports containing package risk, metric outputs, and reviewer comments.
- Store assessments in a database for future viewing.
- User authentication and privilege management.

Youtube video

YouTube video player interface showing a video titled "Why create a Shiny app?" by Appsilon.

The video content includes a slide titled "Why create a Shiny app?" with the following points:

- Main goal: complement GxP "package inclusion requests" with data.
- Shifts responsibility of assessing package risk to the person(s) requesting it
- Leverage the app's risk-based "decision triage" (based on org-defined rules) to output a risk summary report
- Changes communication from "Can we add this?" to a informed "We can add this."

The slide also shows a diagram of a risk assessment process and a Shiny app interface.

Video player controls show the video is at 10:01 / 1:00:28. The video is part of a series titled "Introduction & Welcome".

Below the video player, the video title is displayed: "ShinyGatheringxPharmaverse Workshop: Evaluate Package Quality with the {riskassessment} application".

The video is uploaded by Appsilon, with 39,410 subscribers. The channel name is "チャンネル登録".

Engagement icons show 5 likes, 0 comments, and 0 shares.

On the right side, a list of recommended videos is shown:

- Shiny Gatherings x Pharmaverse: Building Clinical... (Appsilon, 876 回視聴, 4 か月前)
- Clinical submissions with admiral: an R-based ADaM... (Appsilon, 442 回視聴, 7 か月前)
- How to Use DeepSeek R1 for Free in Visual Studio Code... (Chao Charles, 1757 回視聴, 20 時間前)
- From Code to Compliance: Building Validated Shiny Apps... (Appsilon, 133 回視聴, 1 か月前)
- Untangling Biology: The Power of Data Analysis Pipeline with... (Appsilon, 270 回視聴, 6 か月前)



References

Information:

<https://www.pharmar.org/riskassessment/>

<https://pharmar.github.io/riskassessment/articles/riskassessment.html>

<https://www.youtube.com/watch?v=L0ZZHPh2ec4>

Demo application:

<https://app.pharmar.org/riskassessment/>

Version: 3.1.2 (as of OCT2025)



How to run @ local environment

```
# To install riskassessment from GitHub
```

```
install.packages("devtools")
```

```
library(devtools)
```

```
# Install riskassessment
```

```
devtools::install_github("pharmaR/riskassessment")
```

```
library(riskassessment)
```

```
run_app()
```



Demo (1/12)

A screenshot of a web browser displaying the login page of a "RISK ASSESSMENT APPLICATION". The page has a dark blue background with white text. The title "RISK ASSESSMENT APPLICATION" is centered at the top, followed by the version "**VERSION 3.1.2**" and the instruction "PLEASE AUTHENTICATE". Below this, there are two input fields: "Username :" with the value "ADMIN" and "Password :" with masked characters "*****". A "LOGIN" button is positioned below the password field. At the bottom, a note provides first-time login instructions: "NOTE: TO LOG IN FOR THE FIRST TIME, USE THE ADMIN USER: ADMIN WITH PASSWORD QWERTY1." The browser's address bar shows "http://127.0.0.1:5353" and a "Publish" button is visible in the top right corner of the application window.

http://127.0.0.1:5353

Open in Browser

Publish

RISK ASSESSMENT APPLICATION

****VERSION 3.1.2****

PLEASE AUTHENTICATE

Username :

ADMIN

Password :

LOGIN

NOTE: TO LOG IN FOR THE FIRST TIME, USE THE
ADMIN USER: ADMIN WITH PASSWORD QWERTY1.



Demo (2/12)

Main Page

Shiny app interface for R Package Risk Assessment.

Navigation: Risk Assessment, Database, About, Administrative Tools

R PACKAGE RISK ASSESSMENT APP

USER GUIDE

Buttons: Upload Package, Package Metrics, Source Explorer, Build Report

PACKAGE CONTROL PANEL

PACKAGE NAME:

DATE UPLOADED: PKG VERSION:

STATUS: Please select a package

METRIC RISK: Please select a package

SELECT OVERALL DECISION:

Type Package Name(s):

Or Upload a CSV file: No file sel

Remove Package(s):

[View Sample Dataset](#)

DECISION AUTOMATION:

	METRIC	CONDITIONAL	DECISION
Rule 1	Risk Score	$0 \leq x \text{ \& } x \leq 0.25$	Low Risk
Rule 2	Risk Score	$0.25 \leq x \text{ \& } x \leq 0.7$	Medium Risk

Checkout the app's code!



Demo (3/12)

~test_project - Shiny

http://127.0.0.1:4795 Open in Browser

Risk Assessment

Low Risk Medium Risk High Risk

#9CF94FF #FFD070FF #FF7658FF

APPLY COLORS

PACKAGE CONT

PACKAGE NAME

DATE UPLOADED

STATUS

Please select a package

SELECT OVERALL DE

Decision Automation

By Package Risk Score

☒ Low Risk ☒ Medium Risk ☒ High Risk

Low Risk

0.25

Medium Risk

0.75

High Risk

0.7

Customize risk decision rules

Build Report

USER GUIDE

DECISION

Low Risk

Medium Risk

+



Demo (4/12)

Type or select the package name

Then push >

~host_project - Shiny
http://127.0.0.1:4798 Open in Browser Publish

Risk Assessment Database About Administrative Tools

ASSESSMENT APP

USER GUIDE

PACKAGE CONTROL PANEL

PACKAGE NAME
-

DATE UPLOADED
-

PKG VERSION
-

STATUS
Please select a package

METRIC RISK
Please select a package

SELECT OVERALL DECISION

Type Package Name(s)

- abglasso
- ABHgenotypeR
- abima
- abind
- abjData
- abjutils
- ablasso

Or Upload a CSV file

BROWSE... No file sel

[View Sample Dataset](#)

DECISION AUTOMATION:

	METRIC	CONDITIONAL	DECISION
Rule 1	Risk Score	$- 0 \leq x \leq 0.25$	Low Risk
Rule 2	Risk Score	$- 0.25 \leq x \leq 0.7$	Medium Risk

Checkout the app's code!



Demo (5/12)

~/test_project - Shiny
http://127.0.0.1:4242/ Open in Browser Publish

Risk Assessment Database About Administrative Tools

R PACKAGE RISK ASSESSMENT APP

PACKAGE CONTROL PANEL

PACKAGE NAME

tern

-

tern

admiral

2023-01-17

Upload Package Package Metrics Source Explorer Build Report

Type Package Name(s)

Remove Package(s)

Or Upload a CSV file

BROWSE... No file sel

View Sample Dataset

USER GUIDE

Can select assessed packages

REVIEWED 0.36

SELECT OVERALL DECISION

Low Risk Medium Risk High Risk

INFORMATION:

	METRIC	CONDITIONAL	DECISION
Rule 1	Risk Score	$\sim 0 \leq x \& x \leq 0.25$	Low Risk
Rule 2	Risk Score	$\sim 0.25 \leq x \& x \leq 0.7$	Medium Risk

Checkout the app's code!



Demo (6/12)

~test_project - Shiny

http://127.0.0.1:4798 Open in Browser

Risk Assessment Database About Administrative Tools

R PACKAGE RISK ASSESSMENT APP

Upload Package Package Metrics Source Explorer Build Report USER GUIDE

Tab for metrics

PACKAGE CONTROL PANEL

PACKAGE NAME: tern

DATE UPLOADED: 2025-01-17

PKG VERSION: 0.9.7

STATUS: REVIEWED

METRIC RISK: 0.36

SELECT OVERALL DECISION: Low Risk Medium Risk High Risk

MAINTENANCE METRICS

- MAINTENANCE METRICS
- COMMUNITY USAGE METRICS
- PACKAGE DEPENDENCIES

VIGNETTES

0 ✓ 25

Number of vignettes

NEWS CURRENT

1 ✓ Yes

NEWS contains current version

REPORT BUGS

0 ✓ Yes

URL to report bugs exists

WEBSITE

0 ✓ <https://www.phuse.org/newsroom>

Package public website

MAINTAIN R

0 ✓ lee.zhu@phuse.org

Package maintainer

Checkout the app's code!



Demo (7/12)

~test_project - Shiny

http://127.0.0.1:4798 Open in Browser

Risk Assessment Database

R PACKAGE RISK ASSESSMENT APP

Upload Package Package Metrics **Source Explorer** Build Report

USER GUIDE

PACKAGE CONTROL PANEL

PACKAGE NAME

tern

DATE UPLOADED

2025-01-17

PKG VERSION

0.9.7

STATUS

REVIEWED

METRIC RISK

0.36

SELECT OVERALL DECISION

Low Risk Medium Risk High Risk

Exported Function

to_string_matrix

File Type

R Source Code

utils_rtables.R

FUNCTION EXPLORER

PACKAGE BROWSER

FUNCTION EXPLORER

```
1 # Convert table into matrix of strings
2 #
3 # @description "r lifecycle::badge("stable")"
4 #
5 # Helper function to use mostly within tests. 'with_spaces' pa
6 # to test not only for content but also indentation and table
7 # 'print_txt_to_copy' instead facilitate the testing developm
8 # formatted text that needs only to be copied and pasted in t
9 #
10 # @inheritParams formatters::toString
11 #
12 # @param x ('VTableTree')\n r 'rtables' table object.
13 # @param with_spaces ('flag')\n whether the tested table sho
14 # @param print_txt_to_copy ('flag')\n utility to have a way
15 # into the expected variable instead of copying it too manu
16 #
17 #
18 # @return A 'matrix' of 'string's. If 'print_txt_to_copy'
```

Checkout the app's code! 🐙



Demo (8/12)

Tab for report

PACKAGE CONTROL PANEL

PACKAGE NAME
admiral

DATE UPLOADED
2025-01-17

PKG VERSION
1.2.0

STATUS
REVIEWED

METRIC RISK
0.36

SELECT OVERALL DECISION

BUILD REPORT

REPORT CONFIGURATIONS

Select Format
html
html
docx
pdf

DOWNLOAD REPORT

STORE PREFERENCES

Elements to include:

- ☐ Report Author
- ☐ Report Date
- ☐ Risk Score
- ☐ Overall Comment
- ☐ Package Summary
- ☐ Maintenance Metrics
- ☐ Maintenance Comments
- ☐ Community Usage Metrics
- ☐ Community Usage Comments
- ☐ Package Dependencies
- ☐ Include Suggests
- ☐ Dependency Comments
- ☐ Source Explorer Comments
- ☐ Function Explorer Comments

Check out the app's code!



Demo (9/12)

complot_0.92_Risk_Ass... X 作成

すべてのツール 編集 変換 電子サイン

テストまたはツールの検索

R Package Risk Assessment
Report for Package: complot

Report Date: January 16, 2021

Package:
complot

Version:
0.06

Title:
Visualization of a Correlation Matrix

Description:
Provides a visual exploration tool for correlation matrices that supports automatic variable matching, to help detect variable patterns among variables.

Author:
Jinyan Wei [cre, aut], Vilmar Stuke [wr], Michael Levy [ctb], Yihui Xie [ctb], Yan An [ctb], Jeff Zeng [ctb], Martin Weinand [ctb], Jan Goe [ctb], Thomas Fritzsche [ctb]

Maintainer:
Jinyan Wei <weijinyan@gmail.com>

License:
MIT + file LICENSE

Published:
2021-11-18 12:01:55 UTC by jny

Automatically Generated Date:
2021-08-11

Risk Score:
0.20

Package Decision:
Passing



Demo (10/12)

Database of uploaded packages

~/test_project - Shiny

http://127.0.0.1:7771 Open in Browser Publish

Risk Assessment **Database** About Administrative Tools

UPLOADED PACKAGES



PACKAGE COUNT

2

Number of Packages Uploaded to DB



DECISION COUNT

2 (100%)

Packages with Decisions Made



DECISION SUMMARY

Medium Risk: 2 (100%)

Package Counts by Decision Type

Excel CSV Search:

PACKAGE	DATE UPLOADED	VERSION	SCORE	DECISION	DECISION BY	DECISION DATE	LAST COMMENT	EXPLORE METRICS
tern	2025-01-17	0.9.7	0.36	Medium Risk	Auto Assigned	2025-01-17	2025-01-17 17:50:07.893365	→
admiral	2025-01-17	1.2.0	0.36	Medium Risk	Auto Assigned	2025-01-17	2025-01-17 18:17:15.020344	→

Checkout the app's code!



Demo (11/12)

The screenshot displays the Phuse web application interface. The top navigation bar includes links for Risk Assessment, Database, About, and Administrative Tools. The Administrative Tools section is active, showing sub-options: Credential Manager, Roles & Privileges, and Assessment Reweighting. The main content area is titled 'ADMINISTRATIVE TOOLS & OPTIO' and features a 'USERS' section with a '+ ADD A USER' button. Below this is a table of users with columns for User Name, Start Date, Expiration Date, Admin status, Role, Edit, Remove, and Select. Two users are listed: 'ADMIN' and 'testuser'. A search bar is located above the table. The bottom of the page shows a progress bar and a 'Checkout the app's code!' link.

Administrative Tools

ADMINISTRATIVE TOOLS & OPTIO

Credential Manager Roles & Privileges Assessment Reweighting

USERS

+ ADD A USER

Search:

USER NAME	START DATE	EXPIRATION DATE	ADMIN	ROLE	EDIT	REMOVE	SELECT
ADMIN		2026-01-17	Yes	admin			☐
testuser	2025-01-17	2025-01-17	No	lead			☐

Showing 1 to 2 of 2 entries

Checkout the app's code!



Demo (12/12)

The screenshot shows a web application interface with a sidebar on the left and a main content area. The sidebar has sections for 'PACKAGE', 'PACKAGE NAME', 'DATE UPLOADED', 'STATUS', and 'SELECT OVER'. The main content area shows a file explorer for a project folder named 'test_project'. The file explorer lists files and folders, including 'credentials.sqlite', 'database.sqlite', and 'loggit'. A blue callout box highlights these files and contains the following text:

Database and settings saved under project folder (project where app was run).

When you forget your password, you need to delete these files (uninstalling reassessment might not work).



riskassessment

NOTE:

- risk score in report is NOT the same as pkg score from riskmetric because risk score is calculated from the different number of metrics



val.meter

R package
{val.meter}



Overview

- An experimental rewrite of {riskmetric}

Reference

<https://github.com/pharmaR/val.meter>

Design philosophy

- Precise and reproducible
- Unopinionated
 - By avoiding scores altogether, and instead focusing only on concrete, unopinionated measures of packages



Metrics of {val.meter}

- <https://github.com/pharmaR/val.meter>

Metric	Description
r_cmd_check_error_count	R CMD check Error Count <integer> the number of errors produced when running 'R CMD check'
downloads_total	Total Downloads <integer> total number of lifetime downloads, as reported by the Posit CRAN mirror through the cranlogs API
dependency_count	Dependency Count <integer> the number of required dependencies



Execution of {val.meter}

```
rpkg <- random_pkg(permissions = TRUE)
metrics(rpkg, all = TRUE)
```

```
$name
[1] "wondr"

$version
[1] "2.8.3"

$r_cmd_check_error_count
[1] 1

$downloads_total
[1] 285835

$dependency_count
[1] 0
```



risk.assessr

R package
{risk.assessr}





- risk.assessr helps in the initial determining of a package's reliability and security in terms of maintenance, documentation, and dependencies.
- Users can evaluate package risk as high, medium, or low based on user-defined thresholds.
- The weighting for each risk assessment item can be modified or added, allowing the overall evaluation score to be customized for individual users or companies.
- The project is inspired by the riskmetric package and the mpn.scorecard package and draws on some of their ideas and functions.



risk.assessr - Metrics (1/2) -

Metrics	Description
has_bug_reports_url	Checks if the package provides a URL for reporting bugs.
license	Evaluates the presence and type of license (e.g., MIT, GPL).
has_examples	Checks if functions include usage examples.
has_maintainer	Verifies if maintainer information is available.
has_news	Checks for a NEWS file documenting changes across versions.
size_codebase	Measures the size of the codebase.
has_source_control	Checks if the package uses version control (e.g., Git).
has_vignettes	Verifies the presence of vignettes (long-form documentation).



risk.assessr - Metrics (2/2) -

Metrics	Description
has_website	Checks if the package has a dedicated website.
news_current	Assesses whether the NEWS file is up-to-date.
export_help	Checks if exported functions have help documentation.
export_calc	Analyzes the number and structure of exported functions.
check	Score from R CMD check, which tests package integrity. The score is the weighted sum of notes (0.1), errors (1) and warnings (0.25).
covr	Measures unit test coverage using the covr package.
dep_score	Calculates a score based on the number of dependencies (higher scores indicate greater risk). (Based on 19 dependency packages)
revdep_score	The score is calculated such that it increases with the number of reverse dependencies (Based on 20 reverse dependency packages.).



Risk score and profile

- Weight each item's score, sum them, and subtract that score from 1.

```
13  ✓ create_weights_profile <- function() {  
14      weights <- c(  
15          has_bug_reports_url = .2,  
16          license = .1,  
17          has_examples = .4,  
18          has_maintainer = .2,  
19          has_news = .2,  
20          size_codebase = .2,  
21          has_source_control = .1,  
22          has_vignettes = .4,  
23          has_website = .4,  
24          news_current = .2,  
25          export_help = .2,  
26          export_calc = .2,  
27          check = .75,  
28          covr = .6,  
29          dep_score = .5,  
30          revdep_score = .1  
31      )  
32      return(weights)  
33  }
```

- Scores are classified using the total score. A threshold can be set.

```
8  ✓ create_risk_profile <- function() {  
9  
10      risk_profile <- list(  
11          high_risk_threshold = 1,  
12          medium_risk_threshold = .35,  
13          low_risk_threshold = .25  
14      )  
15      return(risk_profile)  
16  }
```

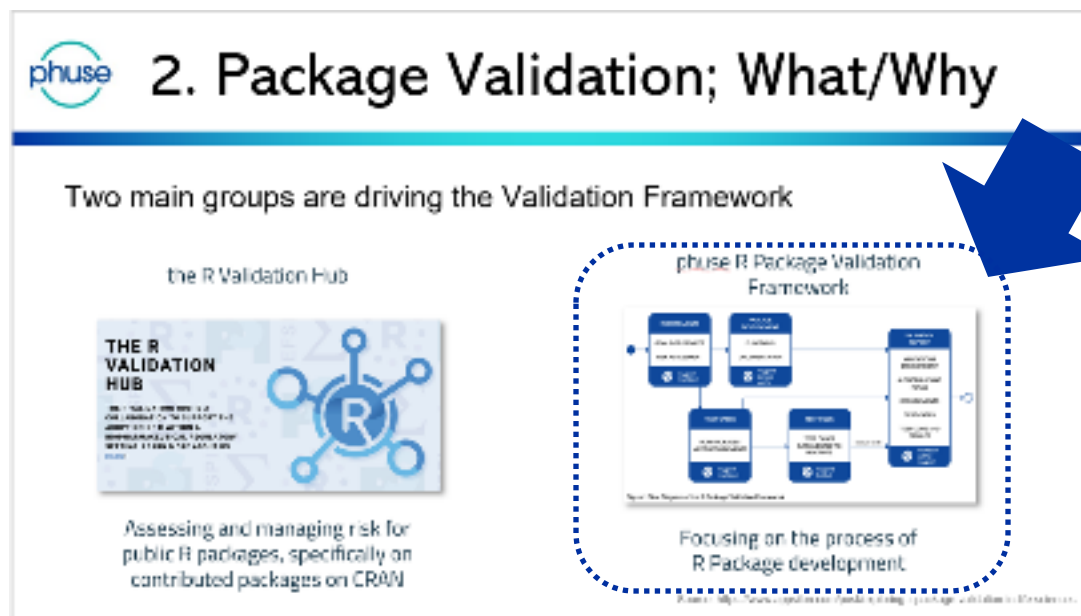


valtools

R package
{valtools}



- <https://phuse-org.github.io/valtools/>
- Support validation tasks of a new packages(creating validation structure), existing packages(bolt-on to existing structure)
- Help flow of requirements, test cases, test codes, documentation

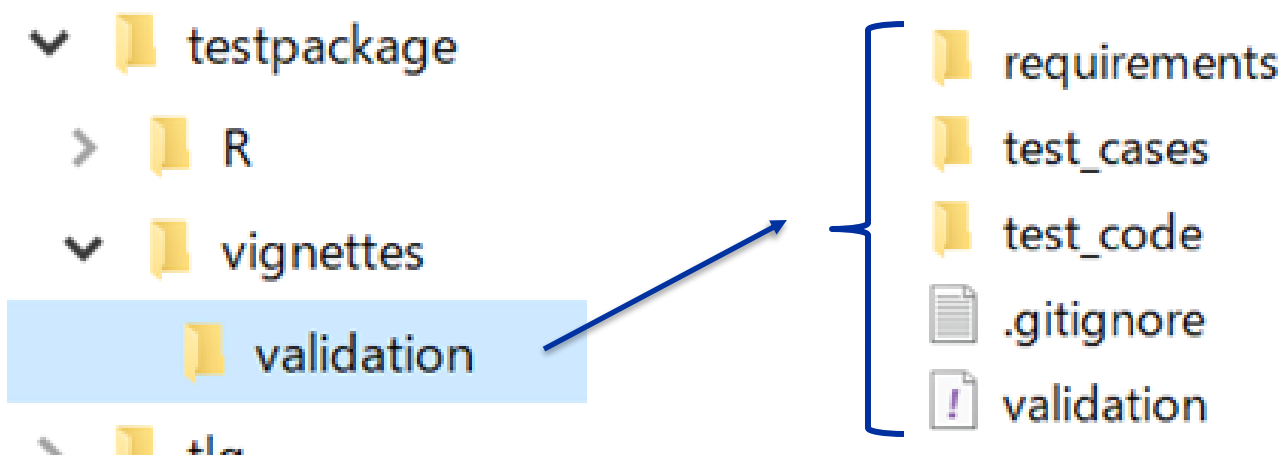


[excerpt] phuse Japan SDE 2024



valtools - sample codes -

```
library(valtools)
usethis::create_package("testpackage") # create a new package folder structure
vt_use_validation(pkg="./testpackage") # create vignettes/validation/validation.yml
vt_use_req("test_requirement")         # create requirement markdown (.md)
vt_use_test_case("test_case1")         # create test case markdown (.md)
vt_use_test_code("test_code1")         # create test code (.r)
setwd("change/workingdirectory/to/testpackage")
vt_use_report()                        # create report R mark down (.rmd)
```





valtools - riskassessment part -

```
#' @title 1.1 login functionality  
#' @editor Ryo  
#' @editDate 2025-11-04  
#' @riskAssessment 1.1 : 4, Medium risk, Moderate Impact  
#' REQUIREMENT: ASSESSMENT
```

In Requirement.md, @riskAssessment is the tag for describing risk level.



valtools - cheat sheet -

valtools :: CHEAT SHEET



Validation Framework

[valtools] implements a convention for organizing information referenced during validation of R packages to generate reports. This framework is consistent for all five (5) validation modes.

Validation Mode	Validation Elements
separate from package	- validation/
from source code	- validation/
at installation	- validation/
after installation	- validation/
for distribution	- validation/

Getting started

`vt_create_package()` Start an R package with validation enabled.

`validateFromSource::create_package()`

`vt_create_gadget()` Initiate just the validation framework. Use when adding to an existing package or validating separate from package.

USERNAMES

[valtools] tracks R package users ("USERS") to quickly tag validation elements with extended details including names for authorship credit, title and role.

`vt_add_user_to_config()` Add additional username details to the configuration file.

`vt_get_all_users()` See a list of usernames already present

Validation Markdown Components

[valtools] populates each component file with some basic info in the header

REQUIREMENTS

`vt_use_req()` initiates and opens a markdown file in the correct folder. Requirements may have additional header info regarding risk assessment.

`@RiskAssessment`

`@ 2.1.1: High Risk, High Impact`

`@ 2.1.2: Medium Risk, Medium Impact`

TEST CASES

`vt_use_test_case()` initiates and opens a markdown file in the correct folder. Test cases may have additional header info regarding coverage, which maps test cases to requirements. A single test case may span multiple requirements.

`@Coverage`

`@ 2.1: 2.1`

`@ 2.2: 2.1, 2.2`

Validation Test Code

`vt_use_test_code()` initiates and opens an R script in the correct folder. Each test is a call to `testthat::test_that()`.

`@ Test that a function works`

`@ Test that a function works`

`@ Test that a function works`

`@ Test that a function works`

Report Code

`vt_use_report/template()` initiate a report from one of the [valtools] templates. The report references content captured in markdown components and test code.

`template = "requirements_adoption"` Only details needed for adoption of requirements.

REPORT ELEMENTS

[valtools] include pointer functions to capture the environment, scraping (rvest) header information and maintaining flexibility across validation modes. Use these to customize a report.

`vt_gather()` Path to the validation folder, used in place of `here::here()`.

`vt_get_child_files()` list of files in markdown component and test code

`vt_file()` - for a sector of markdown component and/or test code files, render as child document in the report. Used in place of setting (font) chunk options "child".

`vt_scrape_*` family of functions to scrape info into data frames:

`vt_scrape_pkg_table()` - usernames and roles

`vt_scrape_val_env()` - environment details

`vt_scrape_change_log()` - changelog details

`vt_scrape_risk_assessment()` - summarize all risk assessment tags in requirements files

`vt_scrape_coverage_matrix()` - summarize all coverage tags in test case files

`vt_scrape_function_editors()` - function editor info

`vt_scrape_requirement_editors()` - requirement editor info

`vt_scrape_test_case_editors()` - test case editor info

`vt_scrape_test_code_editors()` - test code editor info

`vt_kable_*` - converts the output of a `vt_scrape_*` call in a kable object for embedding in report

Validation

Running validation compiles the report code to an eye-readable report suitable for sharing, including evaluation of test code and pointers in the current environment.

separate from package

`vt_validate_report()`

Use only code that has already been installed. Current working directory is used.

from source code

`vt_validate_source()`

Install package code in temp location to render report. Does not allow current work space.

at installation

`vt_validate_install()`

Install package to run report. Persistent updates to `update.packages()`.

after installation

`vt_validate_installed_package()`

Run validation on installed package that was built in code order (valtools) or edgeR.

for distribution

`vt_validate_build()`

Build a package tarball. Sample validation distribution via CRAN. Users will be able to validate after installation.



<https://github.com/phuse-org/valtools/blob/main/man/figures/cheatsheets/valtools-cheatsheet.pdf>



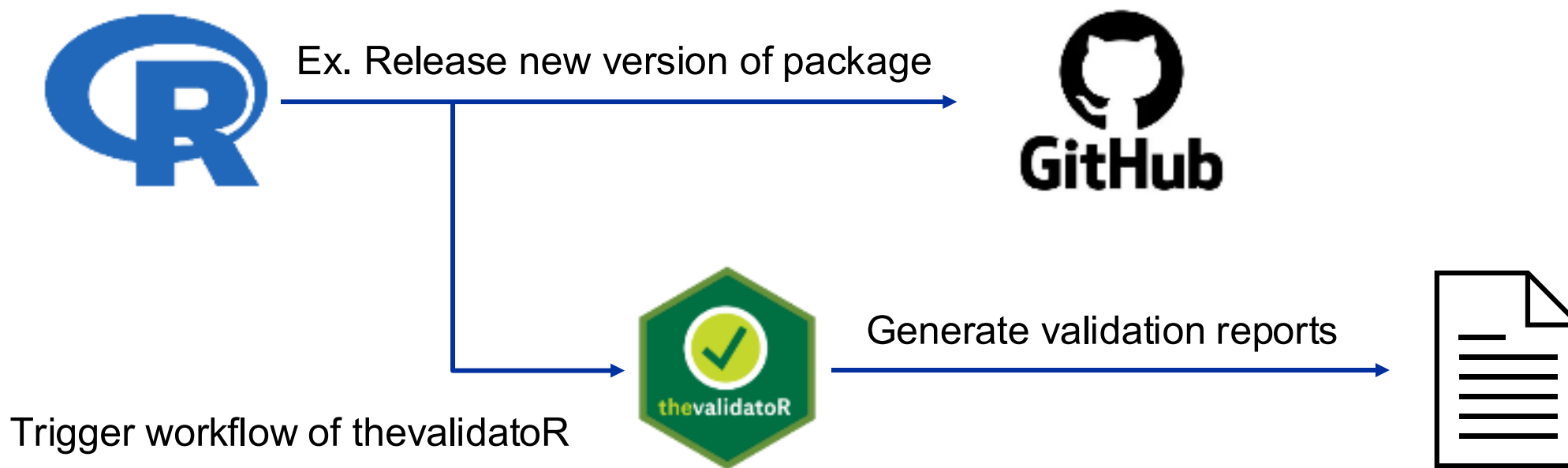
thevalidatoR

GitHub Actions
{thevalidatoR}



Overview

- GitHub Actions workflow that automatically generates validation reports for R packages.
- [insightsengineering/thevalidatoR: Github Action that generates R Package Validation documentation](#)





Validation reports

- Contents
 - Information of environment
 - Result of R CMD Check
 - Unit test coverage checked by covr::package_coverage
 - Link documentation to unit tests generated by {covtracer}

Validation Report	
admiral (v0.4.0)	
Server: https://github.com	Repository: epijim/admiral
Reference: refs/tags/v0.3.0-gh-action-example	
Commit SHA: a63a58d20c8c41ac79849e02a41ff656435e0f0f	
Fri Oct 22 12:02:22 PM 2021	
Contents	
1	Context 1
2	Installation environment and package 1
2.1	System Info 1
2.2	Package installed 2
2.3	R Session Info 2
3	Metric based risk assessment 3
4	Installation documentation 3
4.1	R CMD check 3
4.2	Testing Coverage 5
4.3	Traceability 6



How to use thevalidator

- Create yml file in .github/workflows/
- ex) {rbmi}
- You can use template from [here](#).

```
rbmi / .github / workflows / validate_generic_master.yml
cicloguy for: Use newer R and theValidator versions (#465)
Code Blame 48 lines (50 loc) - 1.01 KB
1 ---
2 name: R Package Validation report
3
4 on: # Run this action when a release is published
5   release:
6     types: [published]
7
8 jobs:
9   r-pkg-validation:
10     name: Create report
11     runs-on: ubuntu-latest
12     container:
13       image: rocker/verse:4.4.2
14     # Set Github token permissions
15     env:
16       GITHUB_PAT: ${ secrets.GITHUB_TOKEN }
17     permissions:
18       contents: write
19       packages: write
20       deployments: write
21     steps:
22     - name: Checkout repo
23       uses: actions/checkout@v4.1.1
24
25     - name: Build report
26       id: validation
27       uses: insightswengineering/thevalidator@v2
28       env:
29         GITHUB_PAT: ${ secrets.GITHUB_TOKEN }
```



Where to save validation reports

- Release page of each version of packages in Github
- ex) {rbmi}

rbmi

main 57 branches 10 tags

Go to file Add file Code

Releases 10

govac Release 1.5.1 (MSD) 2020-07-15 days ago 12M Downloads

File	Description	Time
github	Update Docker Pipeline (MSD)	2 months ago
ascade	Convert stan to suggest (MSD)	last year
R	SDS-convert (MSD)	2 weeks ago
data-csv	Alt-formatter (MSD)	2 months ago
data	modify vignette & data	4 years ago
test	Conversion structure for Bayesian M0 (MSD)	2 months ago
man	Release 1.5.1 (MSD)	5 days ago
new	Rd format (MSD)	6 months ago
tests	MSD test (MSD)	last week
scripts	Release 1.5.1 (MSD)	1 day ago
github	Release 1.5.1 (MSD)	1 month ago
github	added github links to other linked issues	2 years ago
github	Convert stan to suggest (MSD)	last year

About

Reference based multiple imputation R package

highlightsengineering.github.io/rbmi/

Assets 3

Asset	Size	Time
validation-report.pdf	145 KB	5 days ago
Source code (zip)		5 days ago
Source code (tar.gz)		5 days ago

v1.5.1 (Latest)

Bug Fixes

- Updated unit test code to comply with CRANs 10 minute limit on test execution by @grvesti in #528

Contributors



grvesti

Assets 3

validation-report.pdf	145 KB	5 days ago
Source code (zip)		5 days ago
Source code (tar.gz)		5 days ago



Summary of tools (1/2)

Information as of 01Oct2025

	riskmetric	risk.assessr	val.meter
Version	0.2.5 (on CRAN)	2.0.1 (on CRAN)	0.0.0.9000 (on GitHub)
Developed by	R validation Hub (Pharmaverse)	Sanofi (Pharmaverse)	R validation Hub
Main Features	Provides a framework to quantify the risk of using R packages by collecting a wide range of metrics (documentation, unit testing, community engagement, maintainability).	Automates risk assessment of R packages: extract, install, run tests, perform R CMD check, and calculate risk metrics. Also generates traceability matrices linking functions and tests.	Experimental rewrite of <i>riskmetric</i> with S7-based framework. It emphasizes reproducibility, and it separates the scope of scoring and evaluation from the core package itself.



Summary of tools (2/2)

Information as of 01Oct2025

	riskassessment	Valtools	thevalidatoR
Version	3.1.2 (on Github)	0.4.1.9000 (on Github)	2.0.4 (on GitHub)
Developed by	R validation Hub (Pharmaverse)	PHUSE (Pharmaverse)	Roche
Main Features	An R Shiny application built to supplement the riskmetric package in organizational settings. It allows users to review and interpret riskmetric outputs without writing R code, add commentary to metrics, and assign overall risk levels (low/medium/high).	Automates the validation process for R packages, offering templates and helper functions to build a validation framework. It is designed to integrate into the normal R package development workflow, adding validation parts (e.g. requirements, test cases, reports).	A GitHub Action that automates the generation of a validation report for an R package. It is intended to help integrate package validation into CI/CD workflows.



Applications

- Case studies (2022)
 - <https://pharmar.org/categories/case-studies/>
- Other applications (2023, 2025)
 - <https://pharmasug.org/proceedings/2023/SI/PharmaSUG-2023-SI-212.pdf>
 - https://phuse.s3.eu-central-1.amazonaws.com/Archive/2025/Connect/US/Orlando/PAP_OS17.pdf
 - <https://www.atorusresearch.com/openval/>



Key takeaways

- There are ongoing active discussion and development on tools for risk assessment of R packages and validation tools.
- Companies need to follow the latest information and know functionality of these tools to determine their strategy of risk-based risk assessment of R packages
- In addition to conducting the risk assessment, users should also consider how to ensure the appropriateness and accuracy of programming code in line with a risk-based strategy.



References

- riskmetric
<https://pharmar.github.io/riskmetric/>
<https://cran.r-project.org/web/packages/riskmetric/refman/riskmetric.html>
- riskassessment
<https://www.pharmar.org/riskassessment/>
<https://pharmar.github.io/riskassessment/articles/riskassessment.html>
<https://www.youtube.com/watch?v=L0ZZHPh2ec4>
<https://app.pharmar.org/riskassessment/>
- val.meter
<https://github.com/pharmaR/val.meter>
- risk.assessr
<https://sanofi-public.github.io/risk.assessr/>
- Valtools
<https://phuse-org.github.io/valtools/>
- thevalidatoR
[insightsengineering/thevalidatoR: Github Action that generates R Package Validation documentation](https://insightsengineering.com/thevalidatoR/)
- mpn.scorecard
<https://github.com/metrumresearchgroup/mpn.scorecard>

R Package Validation Test investigation

PHUSE Japan OST

Masataka Higashijima (Eli Lilly Japan K.K)
Tomoki Nishikawa (Kyowa Kirin Co., Ltd.)
Kazuyuki Miyashita (Nippon Boehringer Ingelheim Co., Ltd.)
Kensuke Sasaki (Datack Co., Ltd.)

05 Dec. 2025, PHUSE Japan SDE



Agenda

1. Background & Objectives

2. Results

3. Summary

1. Background & Objectives

R Package validation

The process of ensuring that the tools you use for data analysis consistently produce reliable, accurate, and reproducible results.

Accuracy

Ensuring that R packages deliver precise and correct results

Reproducibility

Ensuring consistent analytical outputs across different environments

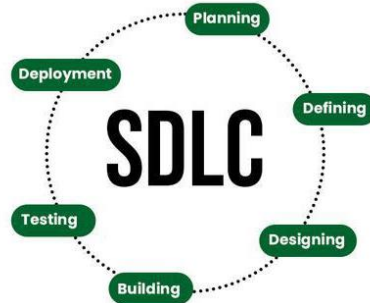
Traceability

Documenting packages and dependencies

1. Background & Objectives

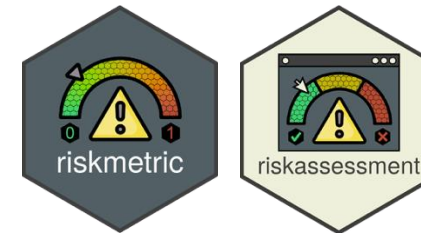
Two approaches for R Package Validation

Software-Based Validation



Validation treats R packages like standard software, demanding solid development practices.

Risk-Based Validation



Focusing on managing risk from third-party/open-source packages.

1. Background & Objectives

Risk-Based Validation



```
# Assess and score R packages
riskmetric::pkg_ref(c("riskmetric", "utils", "tools")) |>
  pkg_assess() |>
  pkg_score()
```

```
> riskmetric::pkg_ref(c("riskmetric", "utils", "tools")) %>%
+   riskmetric::pkg_assess() %>%
+   riskmetric::pkg_score()
# A tibble: 3 × 23
  package version pkg_ref      pkg_score covr_coverage has_news remote_checks news_current
  <chr>    <chr>   <lst_f_p_>    <dbl> <pkg_scor>    <pkg_scor> <pkg_scor>    <pkg_scor>
1 riskmetric 0.2.5 riskmetric<install> 0.493 NA          1         NA          0
2 utils      4.5.1 utils<install>      0.687 NA          0         NA          0
3 tools      4.5.1 tools<install>      0.735 NA          0         NA          0
```

{riskmetric}

- Quantifies the risk associated with using R packages.
- Generates a suite of metrics on package quality and development.
 - maintenance activity
 - community usage
 - test coverage
 - ...
- Tool for package validation and compliance in regulated environments.

1. Background & Objectives

Risk-Based Validation



```
# Assess and score R packages
riskmetric::pkg_ref(c("riskmetric", "utils", "tools")) |>
  pkg_assess() |>
  pkg_score()
```

```
> riskmetric::pkg_ref(c("riskmetric", "utils", "tools")) %>%
+   riskmetric::pkg_assess() %>%
+   riskmetric::pkg_score()
# A tibble: 3 × 23
  package version pkg_ref      pkg_score covr_coverage has_news remote_checks news_current
  <chr>    <chr>   <lst_f_p_>    <dbl> <pkg_scor>    <pkg_scor> <pkg_scor>    <pkg_scor>
1 riskmetric 0.2.5 riskmetric<install> 0.493 NA          1         NA          0
2 utils      4.5.1 utils<install>      0.687 NA          0         NA          0
3 tools      4.5.1 tools<install>      0.735 NA          0         NA          0
```

{riskmetric}

- Quantifies the risk associated with using R packages.
- Generates a suite of metrics on package quality and development.
 - maintenance activity
 - community usage
 - **test coverage**
 - ...
- Tool for package validation and compliance in regulated environments.

1. Background & Objectives

Objectives

To investigate the tests related to R Packages and confirm how validation is performed within the Packages

* Focusing particularly on Accuracy within Package Validation (Accuracy, Reproducibility, Traceability).

1. Background & Objectives

Questions about testing within the R Packages

- 1) **Structure of the tests Directory**
- 2) **How are R packages tested?**
- 3) **Examples of Actual Test Cases**

2. Results



1) Structure of the tests Directory

```
package/  
├── R/  
│   ├── xxx.R # functions  
│   └── ...  
├── DESCRIPTION  
├── tests/  
│   ├── testthat.R # Test Runner  
│   └── testthat/  
│       ├── test_xxx.R # test code  
│       └── ...
```

{testthat}

1. **tests/testthat.R** is invoked as the entry point.
2. All **test-*.R** files located in the **tests/testthat/** directory are executed sequentially.

```
> testthat::test_package("tern")  
Loading required package: rtables  
Loading required package: formatters  
...  
[ FAIL 0 | WARN 0 | SKIP 29 | PASS 2347 ]
```

⌘ Older packages may not use {testthat}. Package structure can vary by design.

2. Results



1) Structure of the tests Directory : {testthat} package

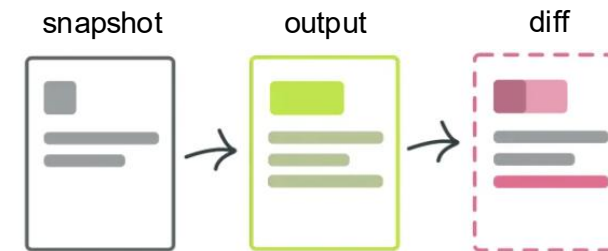
Unit test

- Verifies that individual functions and program units behave exactly as expected
- Test setup, execution, and reporting are covered

Snapshot regression testing

- Automatically compares current outputs (Tables, Plots, Data Frames) against saved “Snapshot” references

```
[ FAIL 0 | WARN 0 | SKIP 29 | PASS 2347 ]
```



2. Results

2) How are R packages tested?



R PACKAGE
DEVELOPER



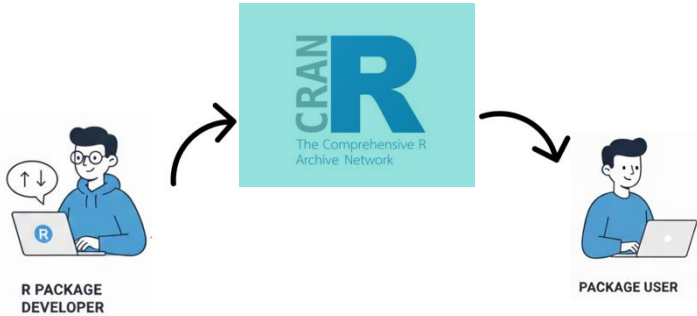
PACKAGE USER

```
# Environment: Developer environment  
# Frequency: On demand  
> devtools::test()
```

```
# Environment: Multiple OS (Linux, macOS, Windows)  
# Frequency: Regularly  
$ R CMD check
```

```
# Environment: User environment  
# Frequency: On demand  
> Install.packages('xxx', type='source', install.tests = TRUE)  
> testthat::test_package('xxx')
```

2. Results



2) How are R packages tested?

CRAN Package Check Results

Last updated on 2025-11-21 07:49:19 CET.

Results for installing and checking packages using the three current flavors of R on systems running Debian GNU/Linux, Fedora, macOS and Windows.

Status summary:

Flavor	OK	NOTE	WARN	ERROR	FAIL	Total	
r-devel-linux-x86_64-debian-clang	16043	6668	164	139	1	23015	Details
r-devel-linux-x86_64-debian-gcc	16093	6657	157	133	3	23043	Details
r-devel-linux-x86_64-fedora-clang	19764	3067	62	137	0	23030	Details
r-devel-linux-x86_64-fedora-gcc	19761	3066	63	144	0	23034	Details
r-devel-windows-x86_64	20449	2390	63	63	3	22968	Details
r-patched-linux-x86_64	20633	2243	19	134	2	23031	Details
r-release-linux-x86_64	20566	2254	26	161	1	23008	Details
r-release-macos-arm64	20799	2069	24	155	0	23047	Details
r-release-macos-x86_64	20911	2026	14	95	0	23046	Details
r-release-windows-x86_64	20768	2132	27	60	3	22990	Details
r-oldrel-macos-arm64	17861	4964	35	176	0	23036	Details
r-oldrel-macos-x86_64	17883	5041	20	92	0	23036	Details
r-oldrel-windows-x86_64	18676	4108	38	157	2	22981	Details

[Results by maintainer](#)

[Results by package](#)

Source: https://cran.r-project.org/web/checks/check_results_tern.html

CRAN Package Check Results																
Last updated on 2025-11-21 07:49:22 CET.																
Results for installing and checking packages using the three current flavors of R on systems running Debian GNU/Linux, Fedora, macOS and Windows.																
Results by maintainer																
Results by package:																
Results with asterisks (*) indicate that checking was not fully performed.																
Package	Version	r-devel Linux x86_64 (Debian Clang)	r-devel Linux x86_64 (Debian GCC)	r-devel Linux x86_64 (Fedora Clang)	r-devel Linux x86_64 (Fedora GCC)	r-devel Windows x86_64	r-oldrel macOS arm64	r-oldrel macOS x86_64	r-oldrel Windows x86_64	r-patched Linux x86_64	r-release Linux x86_64	r-release macOS arm64	r-release macOS x86_64	r-release Windows x86_64	Maintainer	Priority
AalenJohansen	1.0	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Martin Bladt <martinbladt@math.ku.dk>	
aamatch	0.3.7	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Paul Rosenbaum <rosenbaum@wharton.upenn.edu>	
AATools	0.0.3	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Sercan Kalvaci <sercan.kalvaci@plus.ac.at>	
ABACUS	1.0.0	NOTE	NOTE	OK	OK	OK	NOTE	NOTE	NOTE	OK	OK	OK	OK	OK	Miriam Nath <miriam.nath@gmail.com>	
absequence	0.1.0	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Andrew Pilry <sandy.pilry@uky.edu>	
abbreviate	0.1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Sirbert Klinker <ssirbert@hu-berlin.de>	
abc	2.2.2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Blum Michael <michael.blum@temp.at email.com>	
abc.data	1.1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Blum Michael <michael.blum@temp.at email.com>	
ABC.RAP	0.9.0	NOTE	NOTE	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Abdullatif Alshah <alshah@mathematik.uni-muenchen.de>	
ABCAnalysis	1.2.1	NOTE	NOTE	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Florian Lech <lecf@mathematik.uni-muenchen.de>	
ABCscores	6.0.1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	Jamesch Linkersdorfer <edwin.service@gmail.com>	
abclass	0.5.0	OK	OK	OK	OK	OK	NOTE	NOTE	NOTE	OK	OK	OK	OK	OK	Wenjie Wang <wangjie@scs.eupjia.org>	

Source: http://cran.r-project.org/web/checks/check_summary_by_package.html

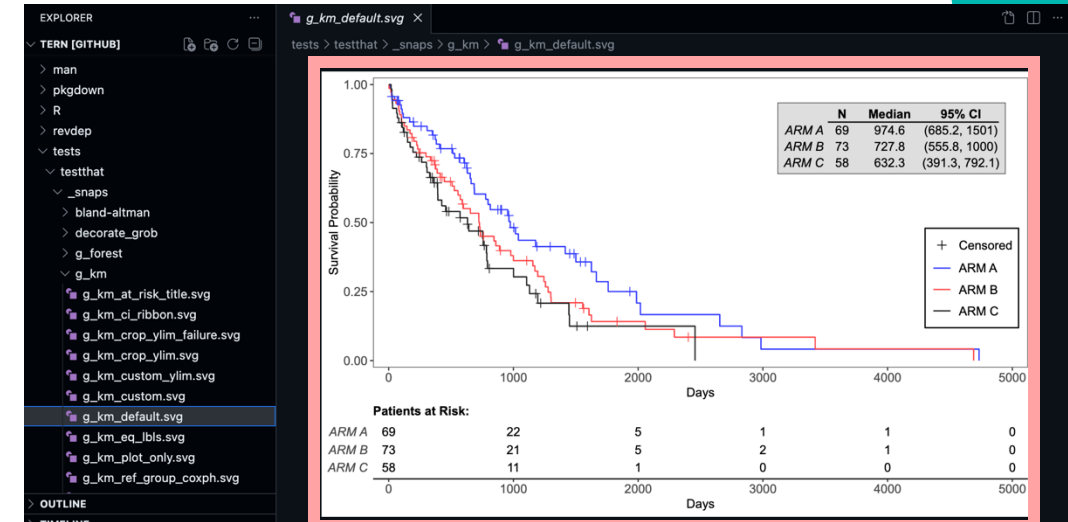
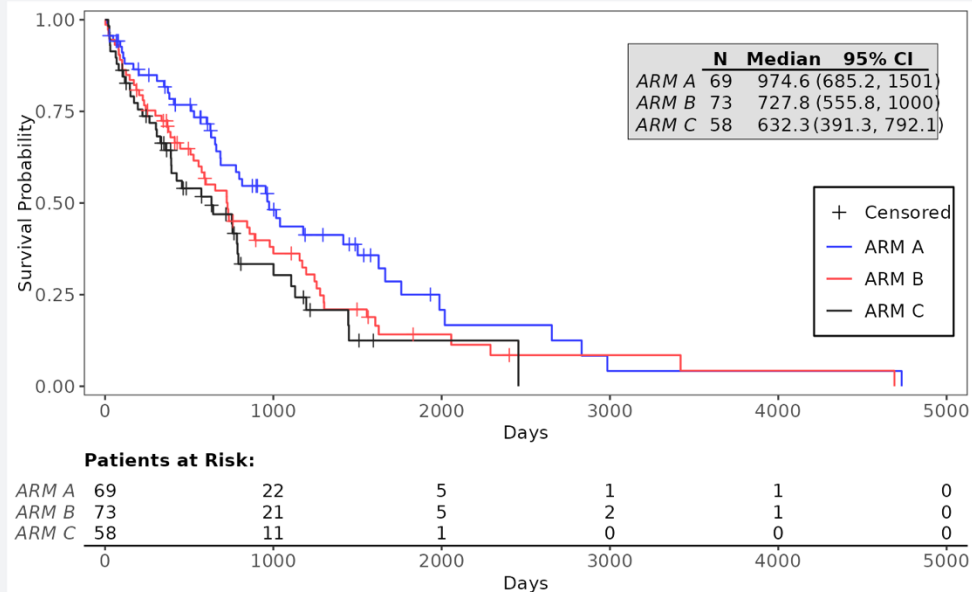
- Checks run regularly across multiple CRAN Package Check Flavors; includes dependency checks.
- Errors/Warnings detected trigger a required contact/fix notice to the Package Maintainer.
- Some tests can be skipped to meet CRAN requirements. If * is marked, it may be better to check carefully.
- Vignettes and Examples code must execute without error

2. Results

3) Examples of Actual Test Cases

`tern::g_km()`

```
# Basic examples
g_km(df = df, variables = variables)
```



Test `g_km()` by comparing with snapshot

```
testthat::test_that("g_km default plot works", {
  testthat::expect_silent(g_km_default <- withr::with_options(
    opts_partial_match_old,
    g_km(
      df = df,
      variables = variables
    )
  ))
  expect_snapshot_ggplot("g_km_default", g_km_default, width = 8, height = 5)
})
```

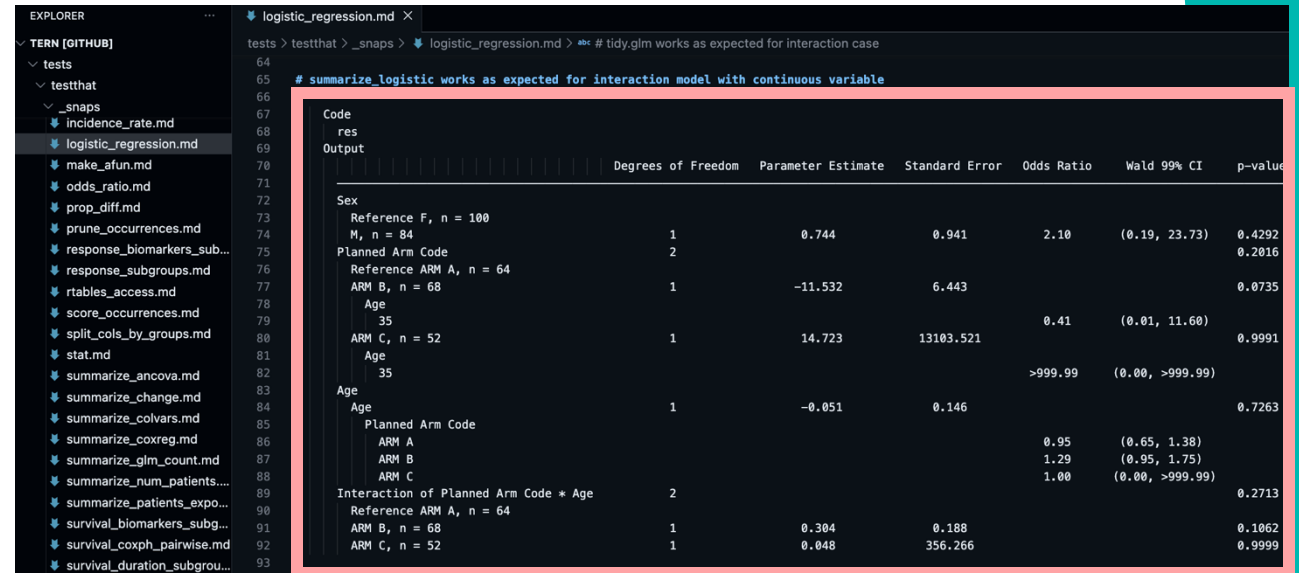
tests/testthat/test-g_km.R

2. Results

3) Examples of Actual Test Cases

`tern::summarize_logistic()`

```
summarize_logistic(  
  lyt,  
  conf_level,  
  drop_and_remove_str = "",  
  .indent_mods = NULL  
)
```



logistic_regression.md

tests > testthat > _snaps > logistic_regression.md > # tidy.glm works as expected for interaction case

summarize_logistic works as expected for interaction model with continuous variable

Code	res	Output	Degrees of Freedom	Parameter Estimate	Standard Error	Odds Ratio	Wald 99% CI	p-value
Sex								
Reference F, n = 100								
M, n = 84			1	0.744	0.941	2.10	(0.19, 23.73)	0.4292
Planned Arm Code			2					0.2016
Reference ARM A, n = 64								
ARM B, n = 68			1	-11.532	6.443			0.0735
Age								
35						0.41	(0.01, 11.60)	
ARM C, n = 52			1	14.723	13103.521			0.9991
Age								
35						>999.99	(0.00, >999.99)	
Age			1	-0.051	0.146			0.7263
Planned Arm Code								
ARM A						0.95	(0.65, 1.38)	
ARM B						1.29	(0.95, 1.75)	
ARM C						1.00	(0.00, >999.99)	
Interaction of Planned Arm Code * Age			2					0.2713
Reference ARM A, n = 64								
ARM B, n = 68			1	0.304	0.188			0.1062
ARM C, n = 52			1	0.048	356.266			0.9999

```
testthat::test_that("summarize_logistic works as expected", {  
  adrs <- adrs_example  
  mod1 <- fit_logistic(  
    adrs,  
    variables = list(  
      response = "Response", arm = "ARMCD",  
      covariates = c("AGE", "SEX"),  
      interaction = "AGE"  
    )  
  )  
  df <- broom::tidy(mod1, conf_level = 0.99) %>%  
    df_explicit_na(na_level = "_")  
  
  result <- basic_table() %>%  
    summarize_logistic(  
      conf_level = 0.99,  
      drop_and_remove_str = "_"  
    ) %>%  
    build_table(df)  
  
  res <- testthat::expect_silent(result)  
  testthat::expect_snapshot(res)  
})
```

tests/testthat/test-logistic_regression.R

Test
**summarize_lo
gistic()** by
comparing with
snapshot
saved as
markdown file.

3. Summary

To investigate the tests related to R Packages and confirm how validation is performed within the Packages

Structure of the tests Directory

- Test codes are located in tests/testthat/, organized to correspond directly with specific functions.

How are R packages tested?

- Packages on CRAN undergo regular, automated checks across multiple environments.
- However, since some tests may be intentionally skipped, users are encouraged to review the test coverage for critical functions manually.

Examples of Actual Test Cases

- Modern packages utilize {testthat} to ensure accuracy.
- Users can verify the correctness of functions by comparing actual outputs with the expected values stored in the _snaps/ directory.

References

- <https://www.appsilon.com/post/r-package-validation-in-pharma>
- <https://r-pkgs.org/>
- <https://cran.r-project.org/web/checks/>
- <https://cran.r-project.org/web/packages/policies.html>
- <https://romannurik.github.io/SlidesCodeHighlighter/>
- <https://github.com/insightsengineering/tern>
- <https://testthat.r-lib.org/>
- <https://gemini.google.com/app>
- <https://github.com/pharmaR/riskmetric>
- <https://www.geeksforgeeks.org/software-engineering/software-development-life-cycle-sdlc/>