CDISC WEBINAR

Modular by Design: The Admiral S
Transparent ADaM Programming.

Revolution ADaM creation across the industry by bringing companies together to develop one harmonized solution

Jia Liu
Analytical Data Scientist, Roche

Agenda


What is Admiral?
1.
Why use Admiral?
2.
Admiral Core Values
3.
Programming Flow Example
4.

What is admiral about?

Admiral is an **open source modularized** toolbox that enables companies and communities to develop ADaM datasets in R.

Think of admiral as a toolbox of modular blocks (toolbox of R functions)

- Each block has a **stand alone purpose** (each function provides a specific functionality)
- Data Scientists can **create their own blocks** (create own R functions)

Constructing an ADaM dataset should become like building out of blocks that are based on admiral modular functions and user-created modular functions.

Why admiral?

Admiral does not exist in a vacuum.

It is the critical engine in the modern clinical data pipeline.

Across the pharmaceutical industry we all face the same challenge when it comes to analysis and creating ADaMdatasets!

- We all work on our own “standard solutions” for ADaMs
- We all face the challenge of a changing and novel data landscape
- New therapeutic areas and analysis concepts
- Individual “blackbox” solutions instead re-use , co-creation and sharing
- We tend to see siloed and hierarchical approaches as more efficient

SDTMv:
Standardized source data.

ADaM (Admiral):
The derivation layer where analysis-ready datasets are built.

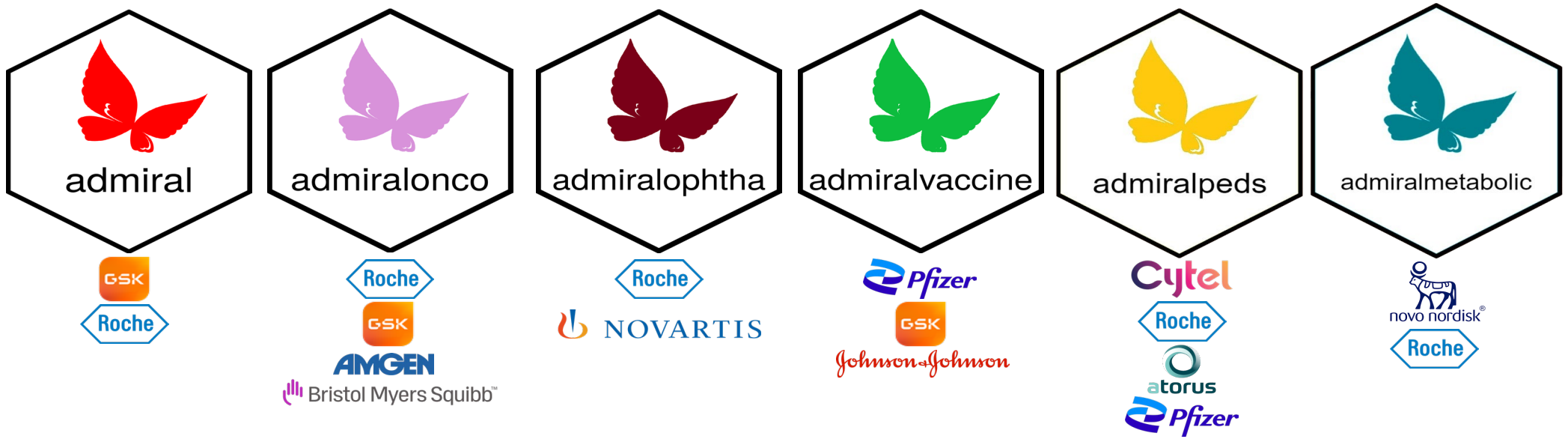
TLG:
The final output for submission.

Moving beyond silos

The industry shift

Historically, every pharma company maintained proprietary, internal SAS macro libraries to do the exact sa
thing.

The New Standard : We now compete on science, but collaborate on utilities. Admiral is co-developed by Roche, GSK, J&J, and others to create a single, robust standard for everyone.

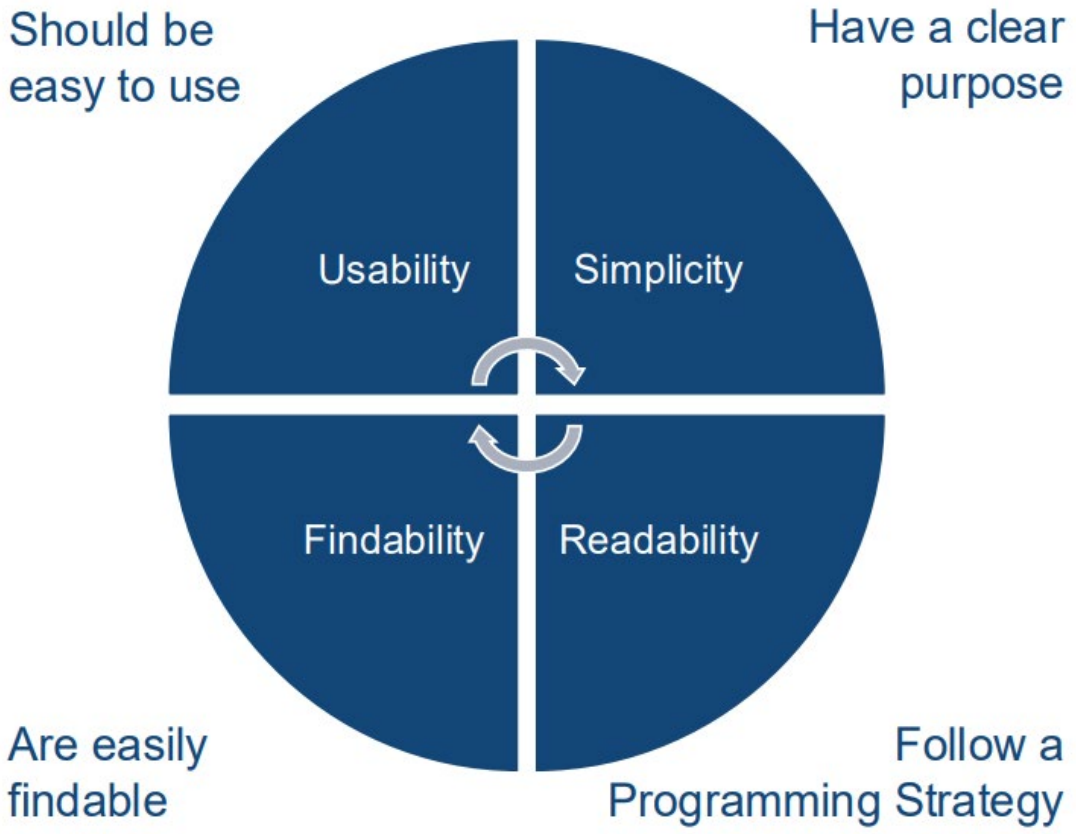


Admiral core values

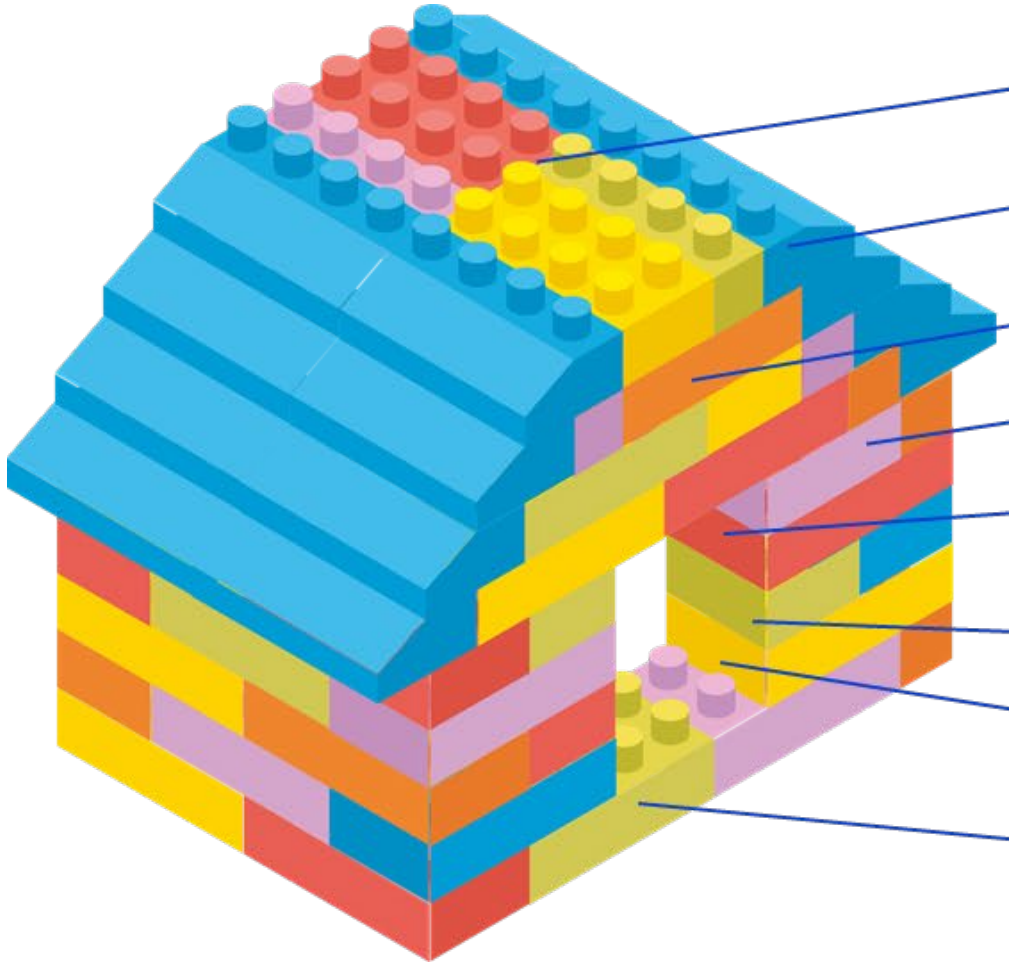
Not a black box, but a modular toolbox

Unlike rigid legacy macros that force an "all-nothing" approach, Admiral functions act like precision tools.

You pick exactly the tool you need for the specific derivation at hand.



Programming flow examples: ADSL



- Add Labels and Attributes
- Derive Other Variables
- Derive Groupings and Populations
- Derive Death Variables (DTHDT, DTHCAUS, LSTALVDT)
- Derive Disposition Variables (EOSDT, DCSREAS, DCSREASP)
- Derive/Impute Date/Time and Duration
- Derive Period, Subperiod, Phase, and Treatment Variables
- Read in Data, load ADSL template

https://pharmaverse.r-universe.dev/articles/admiral/adsl.html

Your essential toolkit

The derivation engine

These 4 functions cover 80% of daily ADaM programming tasks:

- derive_vars_dt():

Character ISO dates (DTC) to numeric (DT).

- derive_vars_merged():

The workhorse for joins (like SQL JOIN).

- derive_vars_merged_lookup():

Maps codes (e.g., VSTESTCD → PARAMCD).

- derive_param_computed():

Creates new parameter rows (e.g., BMI).

The "piping" Paradigm

```
# Readable data flow
adlb %>%
  derive_vars_dt     (DTC) %>%
  derive_vars_dy     (ref_date = TRTSDT) %>%
  derive_anrind     ( ... ) %>%
  add_labels    ()
```

Programming flow examples: ADVS

- Read in Data
- Derive/Impute Numeric Date/Time and Analysis Day (ADT, ADTM, ADY, ADTF, ATMF)
- Assign PARAMCD, PARAM, PARAMN, PARCAT1
- Derive Results (AVAL, AVALC)
- Derive Additional Parameters (e.g. BSA, BMI, or MAP for ADVS)
- Derive Timing Variables (e.g. APHASE, AVISIT, APERIOD)
- Timing Flag Variables (e.g. ONTRTFL)
- Assign Reference Range Indicator (ANRIND)
- Derive Baseline (BASETYPE, ABLFL, BASE, BASEC, BNRIND)
- Derive Change from Baseline (CHG, PCHG)
- Derive Shift (e.g.SHIFT1)
- Derive Analysis Ratio (e.g. R2BASE)
- Derive Analysis Flags (e.g. ANL01FL)
- Derive Criterion Variables (CRITy, CRITyFL, CRITyFN)
- Derive New Rows
- Assign ASEQ
- Add ADSL variables
- Add Labels and Attributes

https://pharmaverse.r-universe.dev/articles/admiral/bds_finding.html

Programming flow examples: ADVS

Read in Data

To start, all data frames needed for the creation of `ADVS` should be read into the environment. This will be a company specific process.

For example purpose, the CDISC Pilot SDTM and ADaM datasets—which are included in `{pharmaversesdtm}`—are used.

```
library(admiral)
library(dplyr, warn.conflicts = FALSE)
library(pharmaversesdtm)
library(lubridate)
library(stringr)
library(tibble)

vs <- pharmaversesdtm::vs
adsl <- admiral::admiral_adsl
vs <- convert_blanks_to_na(vs)
```

Join `ADSL` to `VS` domain.

```
adsl_vars <- exprs(TRTSDT, TRTEDT, TRT

advs <- derive_vars_merged(vs,
  dataset_add = adsl,
  new_vars = adsl_vars,
  by_vars = exprs(STUDYID, USUBJID))
```

USUBJID	VSTESTCD	VSDTC	VISIT	TRTSDT	TRTEDT	TRT01A	TRT01P
01-701-1015	DIABP	2014-01-16	WEEK 2	2014-01-02	2014-07-02	Placebo	Placebo
01-701-1015	DIABP	2014-01-16	WEEK 2	2014-01-02	2014-07-02	Placebo	Placebo
01-701-1015	DIABP	2014-01-16	WEEK 2	2014-01-02	2014-07-02	Placebo	Placebo
01-701-1023	DIABP	2012-08-27	WEEK 2	2012-08-05	2012-09-01	Placebo	Placebo
01-701-1023	DIABP	2012-08-27	WEEK 2	2012-08-05	2012-09-01	Placebo	Placebo
01-701-1023	DIABP	2012-08-27	WEEK 2	2012-08-05	2012-09-01	Placebo	Placebo
01-703-1086	DIABP	2012-09-16	WEEK 2	2012-09-02	2012-12-04	Xanomeline Low Dose	Xanomeline Low Dose
01-703-1086	DIABP	2012-09-16	WEEK 2	2012-09-02	2012-12-04	Xanomeline Low Dose	Xanomeline Low Dose
01-703-1086	DIABP	2012-09-16	WEEK 2	2012-09-02	2012-12-04	Xanomeline Low Dose	Xanomeline Low Dose
01-703-1096	DIABP	2013-02-09	WEEK 2	2013-01-25	2013-03-16	Placebo	Placebo

Programming flow examples: ADVS

Derive/Impute Numeric Date and Analysis Day (`ADT`, `ADY`, `ADTF`, `ATMF`)

1. The function `derive_vars_dt()` can be used to derive `ADT`.

```
advs <- derive_vars_dt(advs,

new_vars_prefix = "A", dtc = VSDTC)
```

2. If imputation is needed and the date is to be imputed to the first of the month, the call would be:

```
advs <- derive_vars_dt(advs,
  new_vars_prefix = "A",
  dtc = VSDTC,
  highest_imputation = "M")
```

3. Once `ADT` is derived, the function `derive_vars_dy()` can be used to derive `ADY`.

```
advs <- derive_vars_dy(advs,
  reference_date = TRTSDT,
  source_vars = exprs(ADT))
```

USUBJID	VISIT	VSDTC	ADT
01-701-1015	SCREENING 1	2013-12-26	2013-12-26
01-701-1015	SCREENING 1	2013-12-26	2013-12-26
01-701-1015	SCREENING 1	2013-12-26	2013-12-26
01-701-1015	SCREENING 2	2013-12-31	2013-12-31
01-701-1015	SCREENING 2	2013-12-31	2013-12-31
01-701-1015	SCREENING 2	2013-12-31	2013-12-31

USUBJID	VISIT	VSDTC	ADT	ADTF
01-716-1024	SCREENING 1	2012-07	2012-07-01	D
01-716-1024	SCREENING 1	2012-07	2012-07-01	D
01-716-1024	SCREENING 1	2012-07	2012-07-01	D

USUBJID	VISIT	ADT	ADY	TRTSDT
01-716-1024	SCREENING 1	2012-07-06	-3	2012-07-09
01-716-1024	SCREENING 1	2012-07-06	-3	2012-07-09
01-716-1024	SCREENING 1	2012-07-06	-3	2012-07-09
01-716-1024	SCREENING 2	2012-07-07	-2	2012-07-09
01-716-1024	SCREENING 2	2012-07-07	-2	2012-07-09
01-716-1024	SCREENING 2	2012-07-07	-2	2012-07-09
01-716-1024	BASELINE	2012-07-09	1	2012-07-09
01-716-1024	BASELINE	2012-07-09	1	2012-07-09
01-716-1024	BASELINE	2012-07-09	1	2012-07-09
01-716-1024	AMBUL ECG PLACEMENT	2012-07-22	14	2012-07-09

Programming flow examples: ADVS

Assign `PARAMCD`, `PARAM`, `PARAMN`, `PARCAT1`

1. To assign parameter level values such as `PARAMCD`, `PARAM`, `PARAMN`, `PARCAT1`, etc., a lookup can be created to join to the source data.

For example, when creating `ADVS`, a lookup based on the SDTM `--TESTCD` value may be created:

2. This lookup may now be joined to the source data:

```
advs <- derive_vars_merged_lookup(
  advs,
  dataset_add = param_lookup,
  new_vars = exprs(PARAMCD),
  by_vars = exprs(VSTESTCD)
)
```

* Please note, it may be necessary to include other variables in the join. For example, perhaps the `PARAMCD` is based on `VSTESTCD` and `VSPOS`, it may be necessary to expand this lookup or create a separate look up for `PARAMCD`.

VSTESTCD	PARAMCD	PARAM	PARAMN	PARCAT1	PARCAT1N
HEIGHT	HEIGHT	Height (cm)	1	Subject Characteristic	1
WEIGHT	WEIGHT	Weight (kg)	2	Subject Characteristic	1
DIABP	DIABP	Diastolic Blood Pressure (mmHg)	3	Vital Sign	2
MAP	MAP	Mean Arterial Pressure	4	Vital Sign	2
PULSE	PULSE	Pulse Rate (beats/min)	5	Vital Sign	2
SYSBP	SYSBP	Systolic Blood Pressure (mmHg)	6	Vital Sign	2
TEMP	TEMP	Temperature (C)	7	Vital Sign	2

USUBJID	VSTESTCD	PARAMCD
01-701-1015	DIABP	DIABP
01-701-1015	HEIGHT	HEIGHT
01-701-1015	PULSE	PULSE
01-701-1015	SYSBP	SYSBP
01-701-1015	TEMP	TEMP
01-701-1015	WEIGHT	WEIGHT
01-701-1023	DIABP	DIABP
01-701-1023	HEIGHT	HEIGHT
01-701-1023	PULSE	PULSE
01-701-1023	SYSBP	SYSBP

Programming flow examples: ADVS

Derive New Rows

Use function `derive_param_computed()` to create a new `PARAMCD`. Below is an example of creating `Mean Arterial Pressure` (`PARAMCD = MAP2`) with an alternative formula.

```
advs <- derive_param_computed(advs,
  by_vars = exprs(USUBJID, VISIT, ATPT),
  parameters = c("SYSBP", "DIABP"),
  set_values_to = exprs(AVAL = (AVAL.SYSBP - AVAL.DIABP) / 3 + AVAL.DIABP,
  PARAMCD = "MAP2", PARAM = "Mean Arterial Pressure 2 (mmHg)"))
```

USUBJID	PARAMCD	VISIT	ATPT	AVAL
01-701-1015	DIABP	AMBUL ECG PLACEMENT	AFTER LYING DOWN FOR 5 MINUTES	67.00000
01-701-1015	MAP2	AMBUL ECG PLACEMENT	AFTER LYING DOWN FOR 5 MINUTES	90.33333
01-701-1015	SYSBP	AMBUL ECG PLACEMENT	AFTER LYING DOWN FOR 5 MINUTES	137.00000
01-701-1015	DIABP	AMBUL ECG PLACEMENT	AFTER STANDING FOR 1 MINUTE	61.00000
01-701-1015	MAP2	AMBUL ECG PLACEMENT	AFTER STANDING FOR 1 MINUTE	83.00000
01-701-1015	SYSBP	AMBUL ECG PLACEMENT	AFTER STANDING FOR 1 MINUTE	127.00000
01-701-1015	DIABP	AMBUL ECG PLACEMENT	AFTER STANDING FOR 3 MINUTES	65.00000
01-701-1015	MAP2	AMBUL ECG PLACEMENT	AFTER STANDING FOR 3 MINUTES	92.00000
01-701-1015	SYSBP	AMBUL ECG PLACEMENT	AFTER STANDING FOR 3 MINUTES	146.00000
01-701-1015	DIABP	AMBUL ECG REMOVAL	AFTER LYING DOWN FOR 5 MINUTES	72.00000

Ready to build?

pharmaverse.github.io/admiral

Your single source of truth for all documentation.

Thank you.

Reference

Admiral R-universe Package
https://pharmaverse.r-universe.dev/admiral

CRAN Release Documentation
https://pharmaverse.github.io/admiral/cran-release/

Admiral GitHub Repository
https://pharmaverse.github.io/admiral/cran-release/

ADSL Guide
https://pharmaverse.r-universe.dev/articles/admiral/adsl.html

BDS Finding Guide
https://pharmaverse.r-universe.dev/articles/admiral/bds_finding

Pharmaverse Slack
https://pharmaverse.r-universe.dev/admiral