

Single Day Event

(ADaM) Dataflow

Xiaochang Liu
Johnson & Johnson



Disclaimer

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

© Janssen Research & Development, LLC. This presentation may not be reproduced, modified, or copied, in whole or in part, without the express written permission of Janssen Research & Development, LLC.

How do data *flow*?

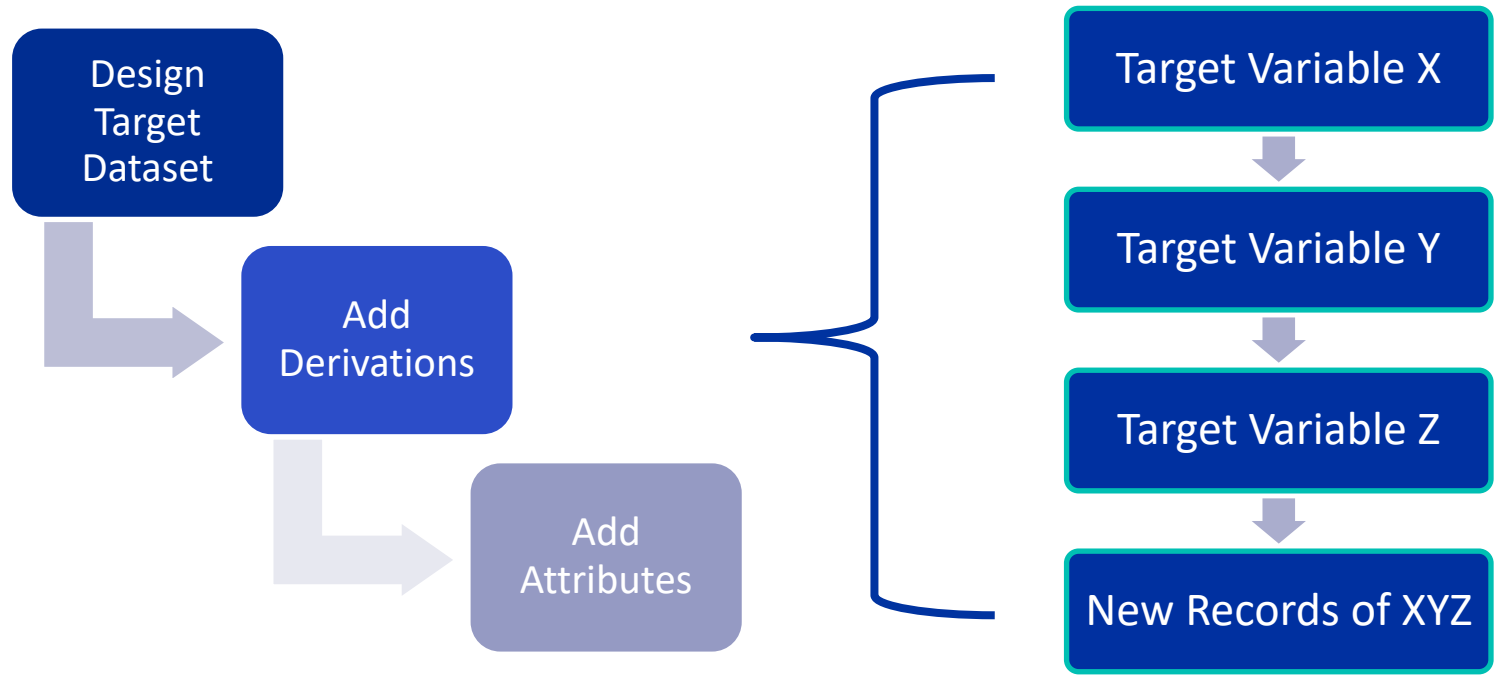


How do data flow? - ETL





How do data flow? - Transformation



Transformation

Close-up of Derivation

Define an ADaM Dataflow



Define an ADaM Dataflow - Code Template

How would you write a SAS/R Script?

- Write everything from scratch?
- Get a “template” from sister studies?
- Comment out a block of code
- Replace a function/macro with a piece of study-specific code



Define an ADaM Dataflow - Code Template

Code "Template" from somewhere

```
# reads sdtm
dm <- haven::read_sas("./dm.sas7bdat")
suppdm <- haven::read_sas("./suppdm.sas7bdat")
ex <- haven::read_sas("./ex.sas7bdat")

# impute time part
ex_imputed <- ex %>%
  impute_this_time(EXSTDTC) %>%
  impute_this_time(EXENDTC)

adsl <- dm %>%
  merge_supp(dm, suppdm) %>%
  derive_general_trtxxp()
```

Modified Code

```
# reads sdtm
dm <- haven::read_sas("./dm.sas7bdat")
# no suppdm in this study
# suppdm <- haven::read_sas("./suppdm.sas7bdat")
ex <- haven::read_sas("./ex.sas7bdat")

# impute time part
ex_imputed <- ex %>%
  impute_this_time(EXSTDTC) %>%
  impute_this_time(EXENDTC)

adsl <- dm %>%
  merge_supp(dm, suppdm) %>%
  # derive_general_trtxxp()
  # study-specific
  mutate(TRT01P = ARM)
```



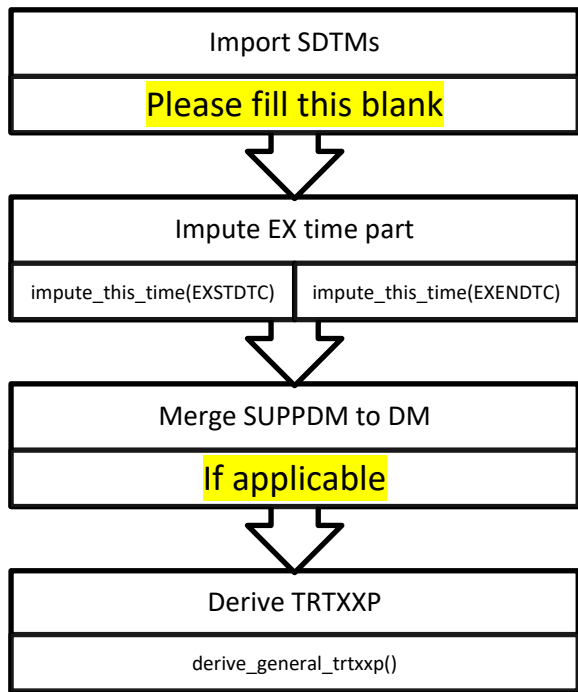
Define an ADaM Dataflow - Code Template

- Is it a **template**?
- Is it **reusable**?
- Are changes **trackable**?
- Can it provide the future programmer with a **clear** guidance?



Define an ADaM Dataflow - Code Template

Fill in the Blanks



Template Script



```
# import SDTM: Please implement

# impute time part
ex_imputed <- ex %>%
  impute_this_time(EXSTDTC) %>%
  impute_this_time(EXENDTC)

adsl <- dm %>%
  # merge suppdm to dm, if applicable
  merge_supp(dm, suppdm) %>%
  # derive TRTXXP
  derive_general_trtxxp()
```



Define an ADaM Dataflow - OOP

Template Script

```
● ● ●  
  
# import SDTM: Please implement  
  
# impute time part  
ex_imputed <- ex %>%  
  impute_this_time(EXSTDTC) %>%  
  impute_this_time(EXENDTC)  
  
adsl <- dm %>%  
  # merge suppdm to dm, if applicable  
  merge_supp(dm, suppdm) %>%  
  # derive TRTXXP  
  derive_general_trtxxp()
```

Convert

To



Object



***Define* an ADaM Dataflow - R Object**

1. Leave blank(s) in the Dataflow
2. Replace implementation and Remain key messages
3. Reuse the same Dataflow for different studies with different implementations
4. Visualize/Inspect a Dataflow in a clearer way



Define an ADaM Dataflow

A Proof of Concept



Define an ADaM Dataflow - OOP

An ADaM dataflow would have 3 typical processes (behaviors)

1. **Importing** data to the workspace
2. **Applying** of derivation(s) to dataset(s)
3. **Exporting** final data to the destination

base_adam

+add_dataset()

+add_derivation()

+export()

Object-oriented Programming (OOP)
of a R6 Class



Define an ADaM Dataflow - OOP

An ADaM dataflow would have 3 typical processes (behaviors)

1. **Importing** data to the workspace
2. **Applying** of derivation(s) to dataset(s)
3. **Exporting** final data to the destination

```
base_adsl$add_dataset(  
  to_ds = "dm",  
  from_ds = "dm.sas7bdat",  
  from_lib = lib  
)  
  
base_adsl$add_dataset(  
  to_ds = "ex",  
  from_ds = "ex.sas7bdat",  
  from_lib = lib  
)  
  
base_adsl$add_derivation(  
  out_ds = "pre_adsl",  
  in_ds = "dm",  
  find_xx_population(ex_dataset = self$get("ex")),  
  derive_racecat(),  
  comments = "derive per SAP"  
)  
  
base_adsl$export()
```

Example code



Define an ADaM Dataflow - Eager vs Lazy

➤ Eager Mode:

Actions that will be executed immediately after you called them.

➤ Lazy Mode:

Actions will be put “on hold” and will not be executed until we want to execute them.

base_adam
+add_dataset() +add_derivation() +export() +define() +run_dataflow()

Object-oriented Programming (OOP)
of a R6 Class



Define an ADaM Dataflow - Eager vs Lazy

➤ Eager Mode:

Actions that will be executed immediately after you called them.



```
base_adsl$add_derivation(  
  out_ds = "pre_adsl",  
  in_ds = "dm",  
  find_xx_population(ex_dataset = self$get("ex")),  
  derive_racecat(),  
  comments = "derive per SAP"  
)
```

Eager Action

➤ Lazy Mode:

Actions will be put “on hold” and will not be executed until we want to execute them.



```
base_adsl$define(  
  add_derivation(  
    out_ds = "pre_adsl",  
    in_ds = "dm",  
    find_xx_population(ex_dataset = self$get("ex")),  
    derive_racecat(),  
    comments = "derive per SAP"  
  )  
)
```

Lazy Action



Define an ADaM Dataflow - Lazy

Why “Lazy” is good?

1. You **don't** need to implement what you can't implement.
2. You may use **trackable** open code.
3. Sister studies may **share** the same workflow.

```
base_adsl$define(  
  not_implemented = TRUE,  
  placeholder = "subset XX population"  
)
```

Leave a Blank to fill

```
base_adsl$implement(  
  add_derivation(  
    out_ds = "dm",  
    in_ds = "dm",  
    filter(grepl("XX", tolower(ACTARM))),  
    comments = "subset XX population"  
  ),  
)
```

Fill the Blank



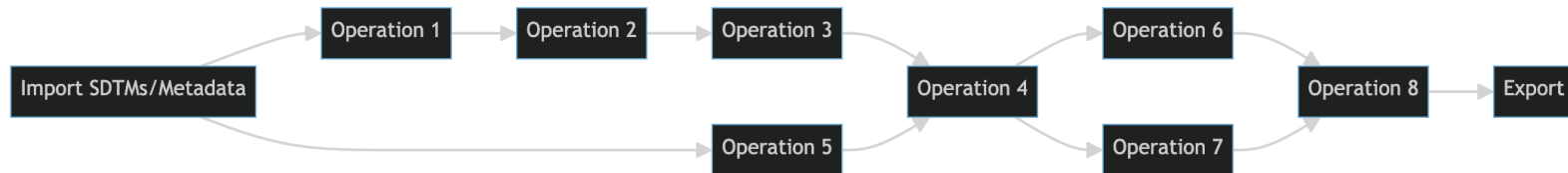
Define an ADaM Dataflow - Answer

- Is it a template? 🙌
- Is it reusable? 🙌
- Are changes trackable? 🙌
- Can it provide the future programmer with a clear guidance? 🤔

Design an ADaM Dataflow



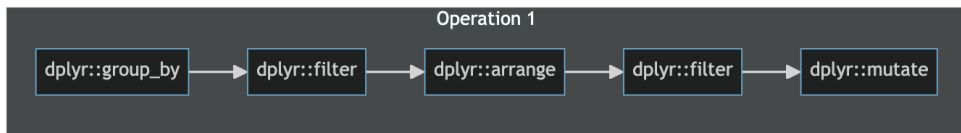
Design an ADaM Dataflow



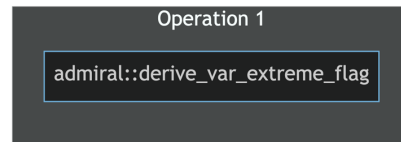
An ADaM *Dataflow* is an *Operation flow*



Design an ADaM Dataflow - Granularity



Option 1: A Series of **basic operations**



Option 2: One **high-level function**

Granularity of a Dataflow is something ***ignored*** by most of the programmers.

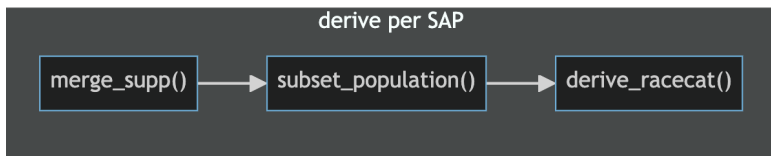
Why consider this factor now?

- help you understand the dataflow
- help you of the future to understand the dataflow
- help newcomers to the project to understand the dataflow

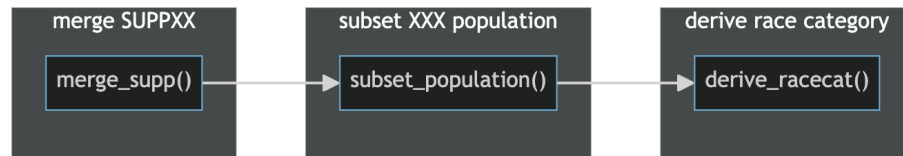


Design an ADaM Dataflow - Granularity

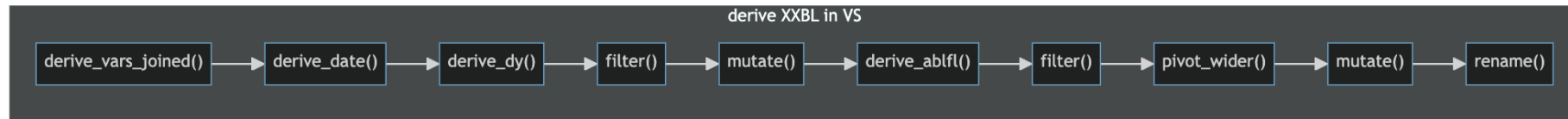
Coarse-grained



Fine-grained



Open code



One More Thing...

***Plug* QC into a Dataflow**



Plug QC into a Dataflow - Why Why Why

Why are there duplicate records in BDS?

Why are there extra values in PARAM compared to sister studies/last data input?

Why are there multiple baselines for the same PARAM in BDS?

Why is the death date in DM not associated with the date from DD?

...

Why is the result of this test in character format?



Plug QC into a Dataflow - The Elephant

“Double programming” is ***de facto*** the “gold standard” approach of quality control of clinical datasets.

- Two programmers implement the same idea and compare each outputs. If two outputs ***match***, then the deliverable is presumed to be correct.
- However, “to ***match***” is also becoming ***de facto*** the aim of most QC programmers.

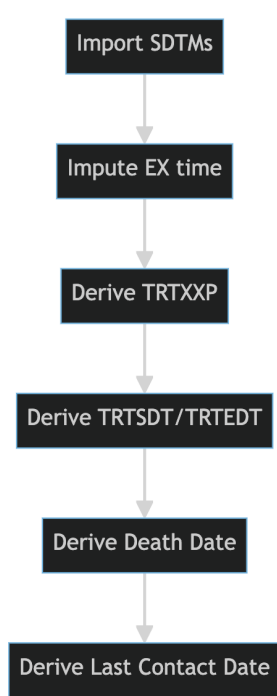


Plug QC into a Dataflow

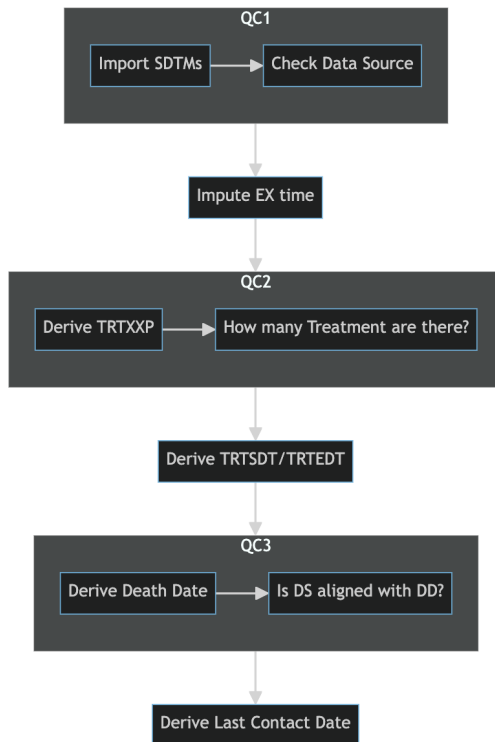
Can the QC process be more target-specific?



Plug QC into a Dataflow - Unit Testing



ADaM Dataflow



ADaM Data/QC Flow

What can be used in a QC block?

- Test Case(s)
- Fit-for-purpose Check(s)
- Double programming...

Why Test Case or Fit-for-purpose Check?

1. Well-described purpose(s)
2. Reusable case(s)/function(s)



Summary

Thank you!



**The Global Healthcare
Data Science Community**

Contact Channels

📞 UK +44 1843 609600
✉ office@phuse.global
🌐 phuse.global

Social Media

🐦 [@phusetwitta](https://twitter.com/phusetwitta)
📘 [/phusebook](https://www.facebook.com/phusebook)
▶ [/phusetube](https://www.youtube.com/channel/UCphusetube)
in [/company/phuse](https://www.linkedin.com/company/phuse)