



File:The Wine MET DP883822.jpg - Wikimedia Commons

Why are old wines sometimes tastier than new ones?

Henning Pontoppidan Föh

Clinical Data Science Director
Novo Nordisk A/S

Lars Lehmann Andersson

Advisory Technical Consultant
SAS Institute

Introduction

Framing of our experience

Programming principles used

Discussion & Conclusion

Prejudice and predisposition



Lars Lehmann Andersson

- Consultant with SAS Institute's Danish branch for 25 years
- Clients in the pharma industry accounts for more than half of the time
- Clients in many other industries, including financial services, energy, and telecommunication
- MSc in engineering



Henning Pontoppidan Föh

- Clinical Data Scientist Director in Biostatistics at Novo Nordisk
- 16+ years pharma industry experience
- Formerly consultant for SAS Institute supporting pharma & various industries
- MSc in physics

DISCLAIMER

Views and opinions expressed are those of the speakers and not necessarily Novo Nordisk or SAS Institute

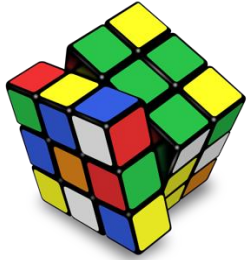
Background – basis of our experience

Factors beyond our control

- Based upon our **experience** back in time 8+ years ago
- Projects **team** consisting of statistical programmer and statistician
 - In-house stat & prog + external consultants
 - Site location: US + EU (DK) + India
- Expectations:
 - **CDISC standards** ← Company standards not yet established
 - **Submission ready**
 - **Scalable** for large Phase 3 trial programme
- This talk is mainly focusing on **ADaM** programming (SDTM handled to large extent by Data Management)
- First experience on Phase 2 → implemented **learnings** into Phase 3 programme

Decisions that helped us

“Choices are the hinges of destiny” – Pythagoras



Right composition of the team

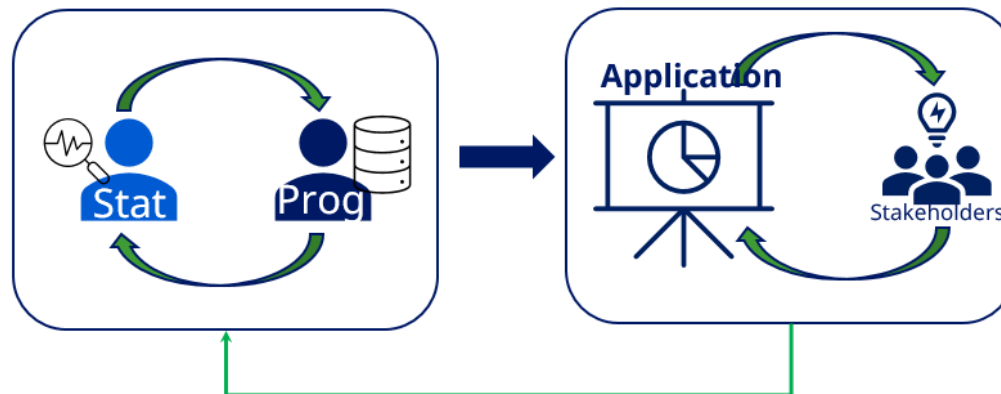
- ADaM ≠ TFL programmers
 - Programming team per trial → copy/paste & adapt to new trials
- Did not hold* → One prog responsible for same dataset across trials!



Ways of working

- Establishment of standards to adhere to: within projects and across project
- Tool for checking consistency

Collaborative development



Programming principles used in the team

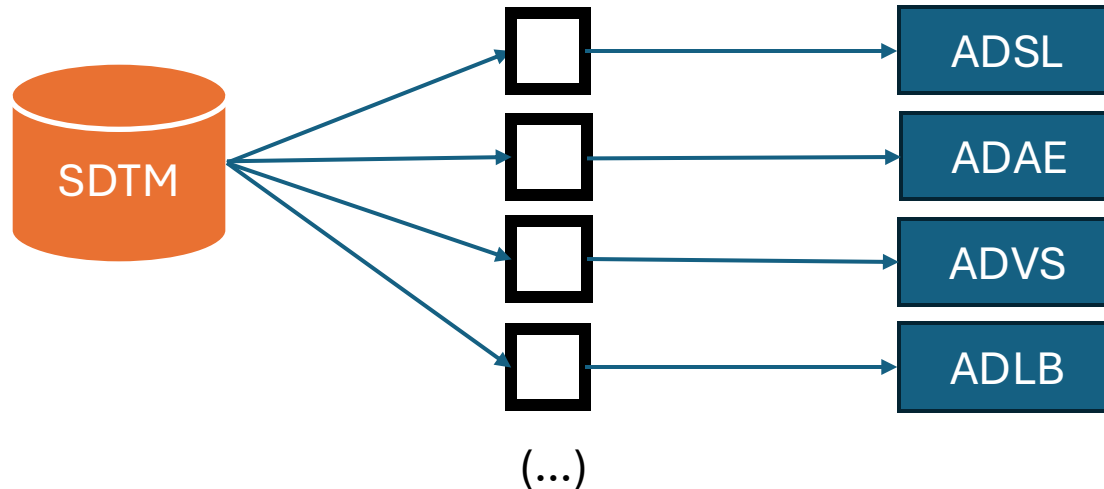
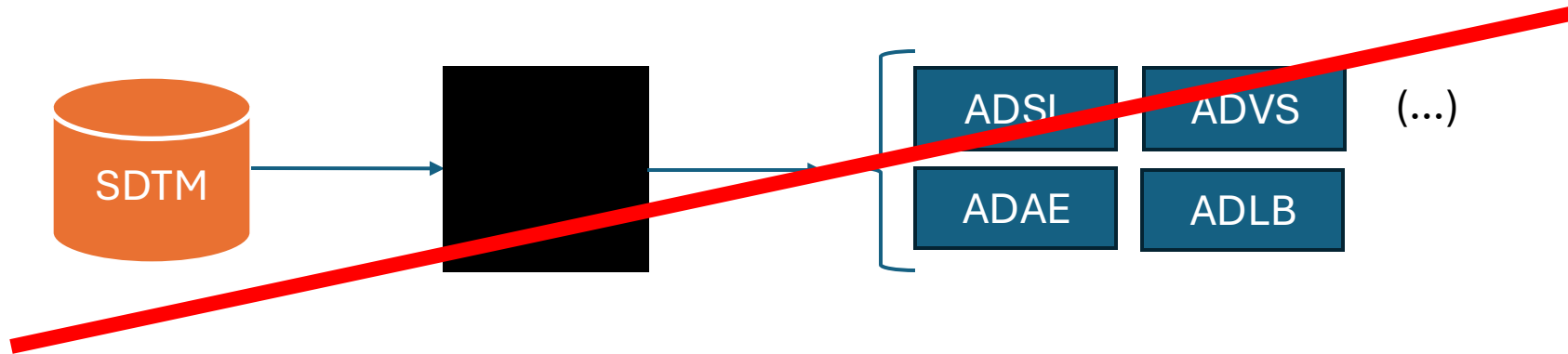
Possibly trivial, but with great outcome. Adherence was key!

Striving for consistency throughout!

- How many programs per ADaM dataset
- Programming the ADaM dataset for the first of many trials
- Centralizing logic – when and how?
- Programs for similar ADaM datasets

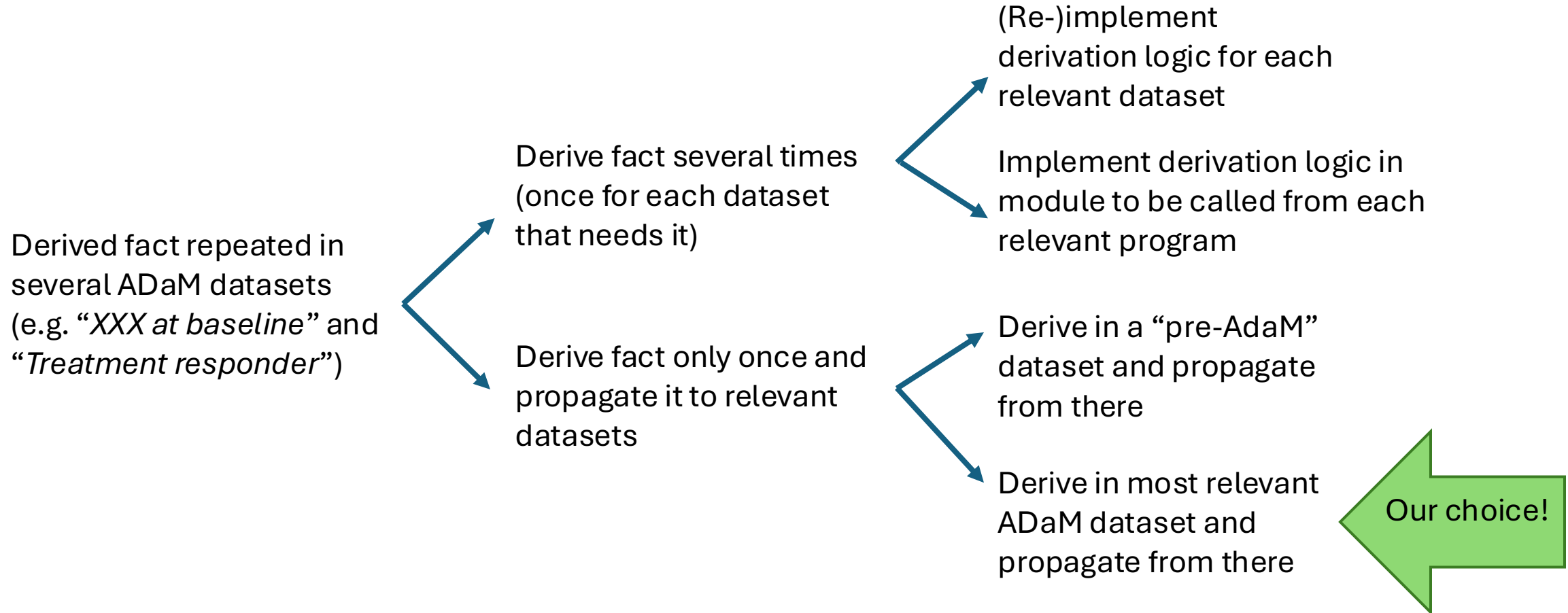
How many programs to make our ADaM datasets?

After all, we want the advantages of it being a team effort



How many programs to make our ADaM datasets?

So... Surely **one** program per ADaM dataset is enough?!

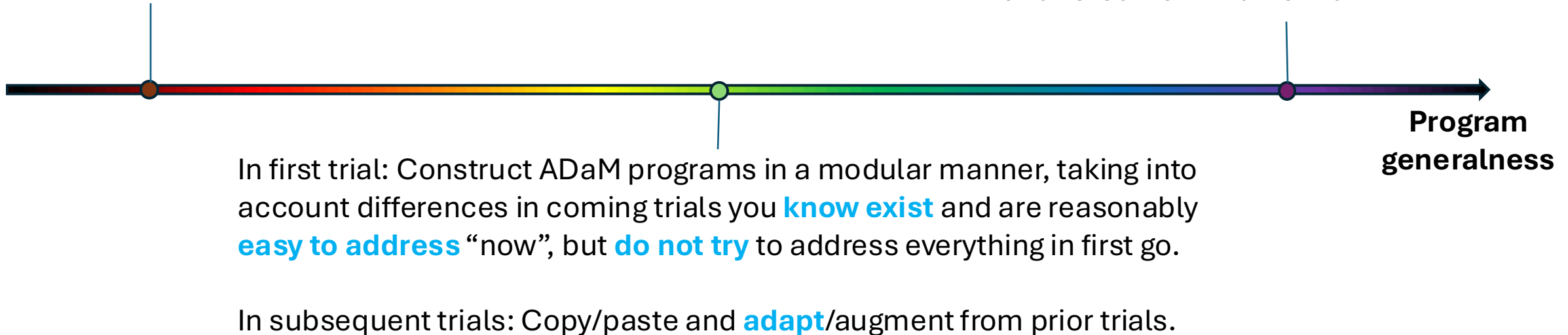


Programming ADaM datasets for first trial in project

How general should a program be?

For each ADaM dataset: Construct a new program more or less from scratch for each of the trials.

For each ADaM dataset: Construct a single program that covers use cases for all trials, so it can be copy/pasted unaltered from trial to trial.



What to put into central logic modules?

And how much in the same module?

We adopted **two** principles for this:

1. If operations A and B often or always occur one after the other in ADaM programs, but **they are not part of the same logical thing** (maybe A is visit reallocation and B is imputation of missing values), then they **should not be part of the same centralized module**. It is preferable to have **more modules that are less complex** over having fewer modules that are more complex.
2. If an operation is so trivial that it is **can be robustly implemented via simple code**, there is **no reason** to embed that simple code in a central module and replace the simple code with a custom module call. A central module should be a **reasonable trade-off** between program legibility and the time involved in developing and maintaining the central module.

Programming similar ADaM datasets

Are all BDS structure datasets the same? And all occurrence datasets?

mk450_adlb.sas

764
765
766 *-----;

mk450_adlb.sas

764
765
766 *-----;

mk450_adlb.sas

634 * Adding columns with baseline value (numeric, character and unit). ;
635 * First, assessments from the visit that for the given topic code is ;
636 * flagged as 'BASELINE' in the business rules are extracted (per ;
637 * subject per topic code), and the value/unit variables are renamed ;
638 * to base etc. ;
639 *-----;

640 proc sql;
641 create table work.step14a as
642 select a.usubjid
643 ,a.topic_cd
644 ,a.avisitn
645 ,a.aval as base
646 ,a.avalc as basec
647 ,a.avalu as baseu
648 from work.step13 as a
649 ,work.baseline_visits_dedup as b
650 where (a.avisitn eq b.blvisitn)
651 and (a.topic_cd eq b.topic_cd)
652 and (a.adelfl eq '')
653 order by 1,2
654 ;
655 quit;
656
657 *-----;
658 * Then we run a deduplication (since for some topic codes, duplicate ;
659 * assessments are allowed), setting the base values to blank if ;
660 * duplicates were found. ;
661 * Finally we left join these deduplicated baseline assessment values ;
662 * onto the lab data. ;
663 *-----;

664 data work.step14b;
665 set work.step14a;
666 by usubjid
667 topic_cd
668 ;
669 if (first.topic_cd) and (last.topic_cd);
670 run;
671
672 proc sql;
673 create table work.step14c as
674 select a.*
675 ,b.base
676 ,b.basec
677 ,b.baseu

mk450_adeg.sas

764
765
766 *-----;

mk450_adeg.sas

634 * Adding columns with baseline value (numeric, character and unit). ;
635 * First, assessments from the visit that for the given topic code is ;
636 * flagged as 'BASELINE' in the business rules are extracted (per ;
637 * subject per topic code), and the value/unit variables are renamed ;
638 * to base etc. ;
639 *-----;

640 proc sql;
641 create table work.step14a as
642 select a.usubjid
643 ,a.topic_cd
644 ,a.avisitn
645 ,a.avalc as basec
646 ,a.avalu as baseu
647 from work.step13 as a
648 ,work.baseline_visits_dedup as b
649 where (a.avisitn eq b.blvisitn)
650 and (a.topic_cd eq b.topic_cd)
651 and (a.adelfl eq '')
652 order by 1,2
653 ;
654 quit;
655
656 *-----;
657 * Then we run a deduplication (since for some topic codes, duplicate ;
658 * assessments are allowed), setting the base values to blank if ;
659 * duplicates were found. ;
660 * Finally we left join these deduplicated baseline assessment values ;
661 * onto the lab data. ;
662 *-----;

663 data work.step14b;
664 set work.step14a;
665 by usubjid
666 topic_cd
667 ;
668 if (first.topic_cd) and (last.topic_cd);
669 run;
670
671 proc sql;
672 create table work.step14c as
673 select a.*
674 ,b.basec
675 ,b.baseu

Compare NavBar

Compare NavBar

Advantages of adhering to the programming principles

Was it really worth it?

- 6 concurrent trials in same project had ADaM programmed in this manner.
- Plenty of programmers involved – little dependency of single individual.
- Copy/paste for easy start, but subsequent adaptations needed.
- Easy to adapt since:
 - Not much “carved in stone”.
 - Implemented in versatile language.
- Many of the principles subsequently rolled out to become company standard at Novo Nordisk.



Discussions

Learnings from our setup

- Our observed value propositions could be programming agnostic
- SAS holds (still) merit for ADaM creation
- SAS should co-exist with other languages that demonstrate strengths in their respective domains
- Decisions to switch should not be driven by “fear of missing out”
- “Fit for purpose” ≠ “one language fits all”

General considerations for change of programming language

- Regulatory Compliance and Validation
- Training and Skills Development
- Interoperability and Integration
- Community and Support
- Innovation and Future-Proofing

Conclusion & summary



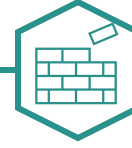
Clear Division of Peoples Work

Assigning specific roles to team members, such as differentiating between individual ADaM and TFL programmers, helped maintain focus and quality



Iterative & Collaborative Development

Engaging statisticians and other stakeholders in iterative development processes, exemplified by the ADTTE dataset creation including a visualization application, fostered better specifications and outcomes



Modular & Adaptable Program Structures

Centralizing shared logic into callable modules and structuring individual ADaM programs according to ETL principles facilitated easier maintenance, QC, and scalability.

Developing flexible program structures that could be easily adapted for new trials ensured efficiency and consistency across similar datasets.



Adherence to Standards

Utilizing a cross-team standards governance board ensured consistency across projects, while internal mechanisms helped quickly rectify deviations from standards.

The background is a close-up, top-down view of a grid of wine bottle bottoms. The bottle caps are dark and have a textured, metallic appearance. The grid is composed of many small squares, each containing a bottle bottom. Overlaid on this grid are several geometric shapes: a large, irregular blue polygon in the center, a white hexagon on the right side, and a white hexagon on the left side. The text "In Vino Veritas..." is centered in the middle of the image.

***In Vino
Veritas...***



Thank you for your attention!

Lars Lehmann Andersson
Lars.L.Andersson@sas.com

Henning Pontoppidan Föh
hpf@novonordisk.com

Questions or comments?