



KNIME Data Talks

*Clinical Analysis Dataset  
Derivation using Visual  
Programming with KNIME*



**Robert Adams** 20220601

Oncology Digitalization & Computational Science

SBU Oncology





# Background – Clinical Data Derivation

A primer into clinical programming and regulatory standards

- Regulatory authorities such as FDA (U.S. Food and Drug Administration) or EMA (European Medicines Agency) require pharma companies to submit their study data in certain standards
- Pharma companies or CROs (clinical research organisations) create study results programmatically usually referred to as “TLFs” (tables, listings, figures)
- A study thereby consists of different derivation levels
  - Usual data flow: (e)CRF / EDC □ RAW □ SDTM □ ADaM □ CSR □ submission
  - The process is more complex by iterations of data cleaning steps and multiple additional standards and guidelines that need to be considered

(e)CRF / EDC – (electronic) Case Report Form / Electronic Data Capture; SDTM – Study Data Tabulation Model; ADaM – Analysis Data Model; CSR – Clinical Study Report



# Background – CDISC (Clinical Data Interchange Standards Consortium)

SDO (standards developing organization)



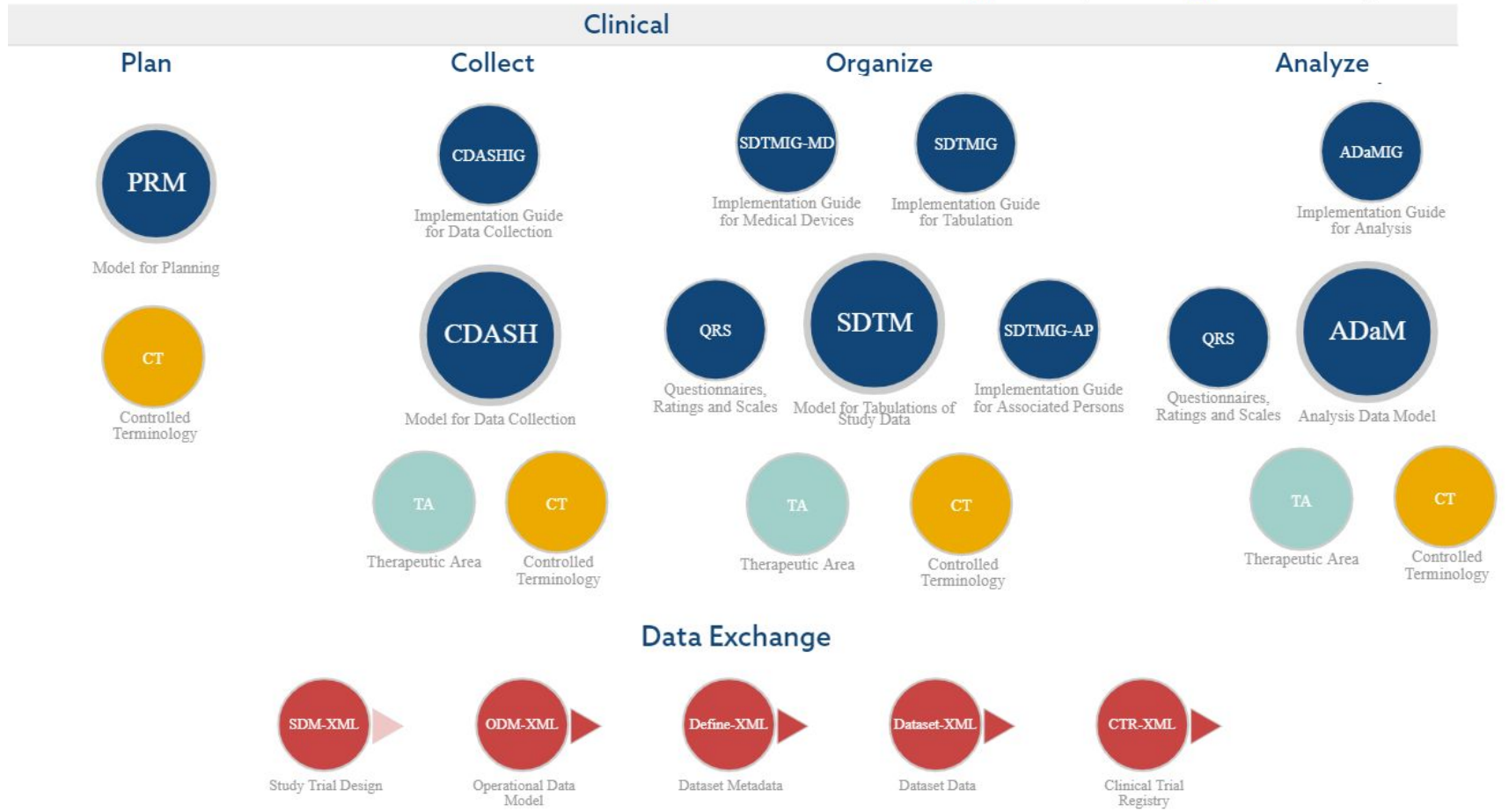
“enable information system interoperability to improve medical research and related areas of healthcare”

<https://www.cdisc.org/standards>

re”  
<https://en.wikipedia.org>

## CDISC Standards in the Clinical Research Process

■ Foundational Standard   ■ Therapeutic Area  
■ Data Exchange   ■ Controlled Terminology





# Background – ADaM (analysis data sets) & Domains

The [CDISC Glossary](#) defines these terms as follows:

- Domain: A collection of logically related observations with a common, specific topic that are normally collected for all subjects in a clinical investigation.

ADaM defines dataset and metadata standards that support:

- efficient generation, replication, and review of clinical trial statistical analyses, and
- traceability among analysis results, analysis data, and data represented in the [Study Data Tabulation Model \(SDTM\)](#).

**ADaM is one of the required standards for data submission to FDA (U.S.) and PMDA (Japan).**

Details on the requirements for FDA are specified in the [FDA's Data Standards Catalog](#) for NDA, ANDA, and certain BLA and the [FDA Guidance on Standardized Data](#).

Details on the requirements for PMDA can be found on the [Advanced Review with Electronic Data Promotion Group p](#)

Dataset	Description
EX	Exposure
DA	Product
ML	DD Dataset Description
PR	EG NV Nervous System Findings
SU	OE SC Subject Characteristics
AE	FT SS Subject Status
BE	GF TR Dataset Description
CE	IE PC TV Trial Visits
DS	IS PE UR Related Records
DV	MH LB VS RELSPEC Related Specimens
HO	CP MB RELSUB Related Subjects
MH	BS LB FA SUPP-- Supplemental Qualifiers for [domain name]
BS	MI RE OI Non-host Organism Identifiers
CP	MB RE TA Trial Arms
CV	MI MK TD Trial Disease Assessments
EC	MS RP TE Trial Elements
	RS TI Trial Inclusion/Exclusion Criteria
	TM Trial Disease Milestones
	TS Trial Summary





# Background – Clinical Programming & Challenges

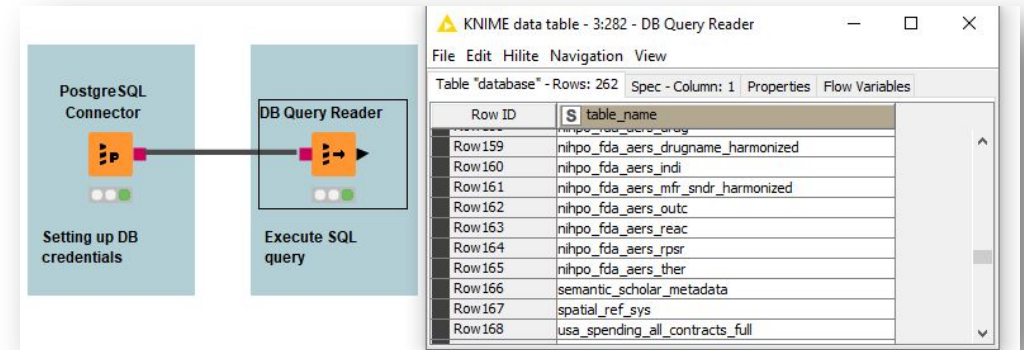


- SAS is a de-facto standard proprietary commercial programming language used by the pharmaceutical industry / regulatory agencies
- TLFs are programmed by SAS experts
- Implementation of CDISC (and other standards) using specific languages follow „SOPs“ (standard operation procedures in companies)
  - Leads to highly interconnected dependencies that makes it challenging to try deviating approaches
- Not all functions involved (e.g. DM – data management – responsible for „clean data“) are necessarily trained programmers
- Although highly standardized, every study is different
  - Maybe 80 % is standardized? 20 % need to be adopted from study to study
    - Especially efficacy domains – different end points defined by (complex) study designs
- Industry dependency for certain software providers
  - No realistic chance in near future to use KNIME for submissions to authorities
  - Currently more of a case study (feasibility) to prove that other technical solutions are possible (spark ideas!) and KNIME could be used for non-regulated (internal) work with clinical data

# Rationale – Why Visual Programming?

An alternative way of deriving clinical analysis data sets

- Double programming / validation alternative
- Intuitive visualization of data flows within the “program“
  - “The program is the documentation”
- Accessibility to new learners
- Standardization
- “Lowest common denominator”
  - Data science concepts shared across different programming & skills backgrounds (e.g. programmers and DM)



- Immediate clarity of algorithmic approach for anyone with programming / data science backgrounds
- Optimization of workflows straight forward



# SAS vs. KNIME

Example:  
Batch  
categorization  
with  
Column  
Expressions

Expression	Type	Collection	Replace Column	Output Column
if (column("DURDIS") >= ...	String	<input type="checkbox"/>	<input type="checkbox"/>	DURDSGR1
column("SITEID")	Number (integer)	<input type="checkbox"/>	<input type="checkbox"/>	SITEGR1

L	DURDIS	S	DURDS...
77			>=12
33			>=12
19			>=12
11			<12
73			>=12
48			>=12
24			>=12

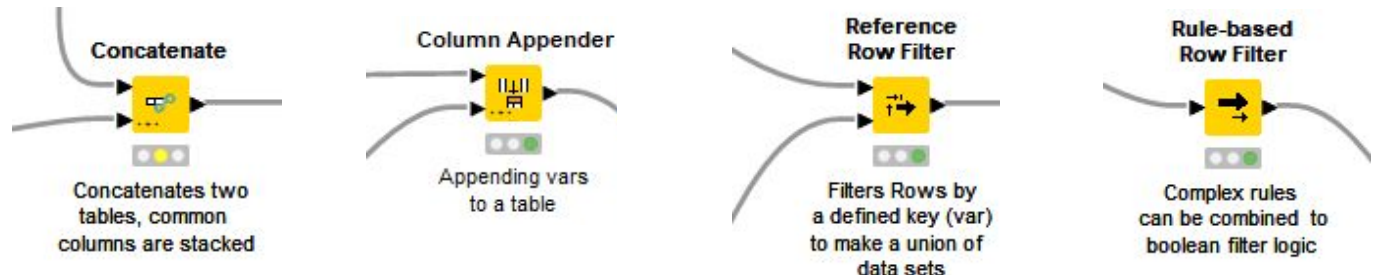
  

```

1 if (column("DURDIS") >= 12){
2   DURDSGR1 = ">=12"
3 } else {
4   DURDSGR1 = "<12"
5 }

```

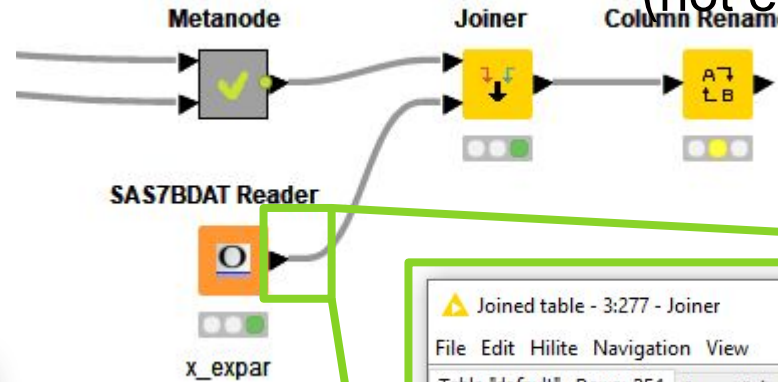
Commonly required ETL  
functionality



Meta node  
(comparable to  
SAS Macros)

Configured  
node  
(not executed)

Data flows



Joined table - 3:277 - Joiner

Table "default" - Rows: 254 Spec - Columns: 10 Properties Flow Variables

Row ID	S	USUBJID	S	ARM	TRTSD...	TRTED...	PARAMN	S
Row0_Row16	01-701-1015	Placebo	02.Jan.2014	02.Jul.2014	6	OVI		
Row1_Row16	01-701-1023	Placebo	05.Aug.2012	01.Sep.2012	6	OVI		
Row2_Row16	01-701-1028	Xanomeline ...	19.Jul.2013	14.Jan.2014	6	OVI		
Row3_Row16	01-701-1033	Xanomeline ...	18.Mär.2014	31.Mär.2014	6	OVI		
Row4_Row16	01-701-1034	Xanomeline ...	01.Jul.2014	30.Dez.2014	6	OVI		
Row5_Row16	01-701-1047	Placebo	12.Feb.2013	09.Mär.2013	6	OVI		
Row6_Row16	01-701-1097	Xanomeline ...	01.Jan.2014	09.Jul.2014	6	OVI		
Row7_Row16	01-701-1111	Xanomeline ...	07.Sep.2012	16.Sep.2012	6	OVI		
Row8_Row16	01-701-1115	Xanomeline ...	30.Nov.2012	23.Jan.2013	6	OVI		
Row9_Row16	01-701-1118	Placebo	12.Mär.2014	09.Sep.2014	6	OVI		

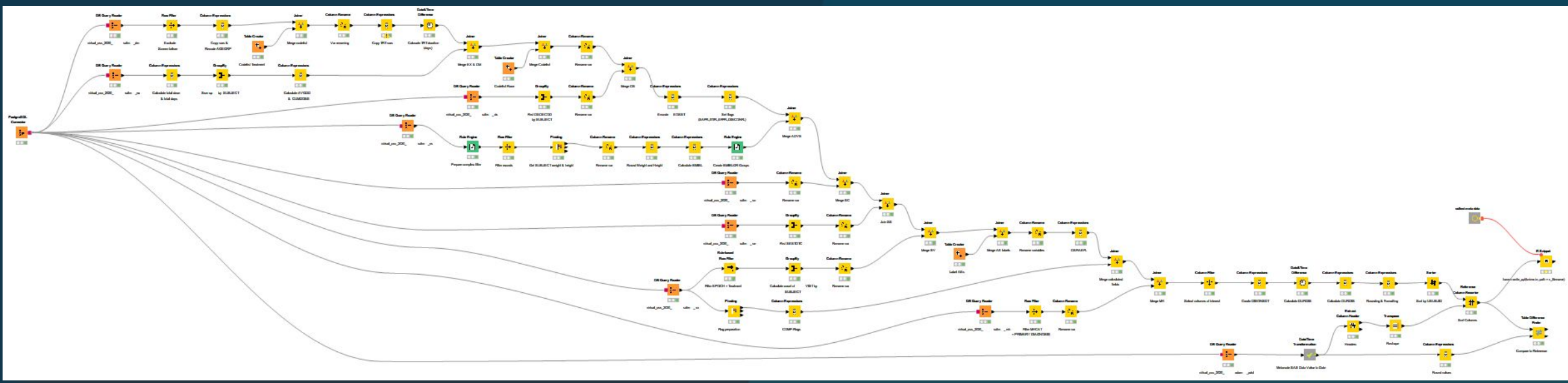


# ADaM Derivation – ADSL (subject listings)

Data retrieval

Derivation

Testing & XPT output





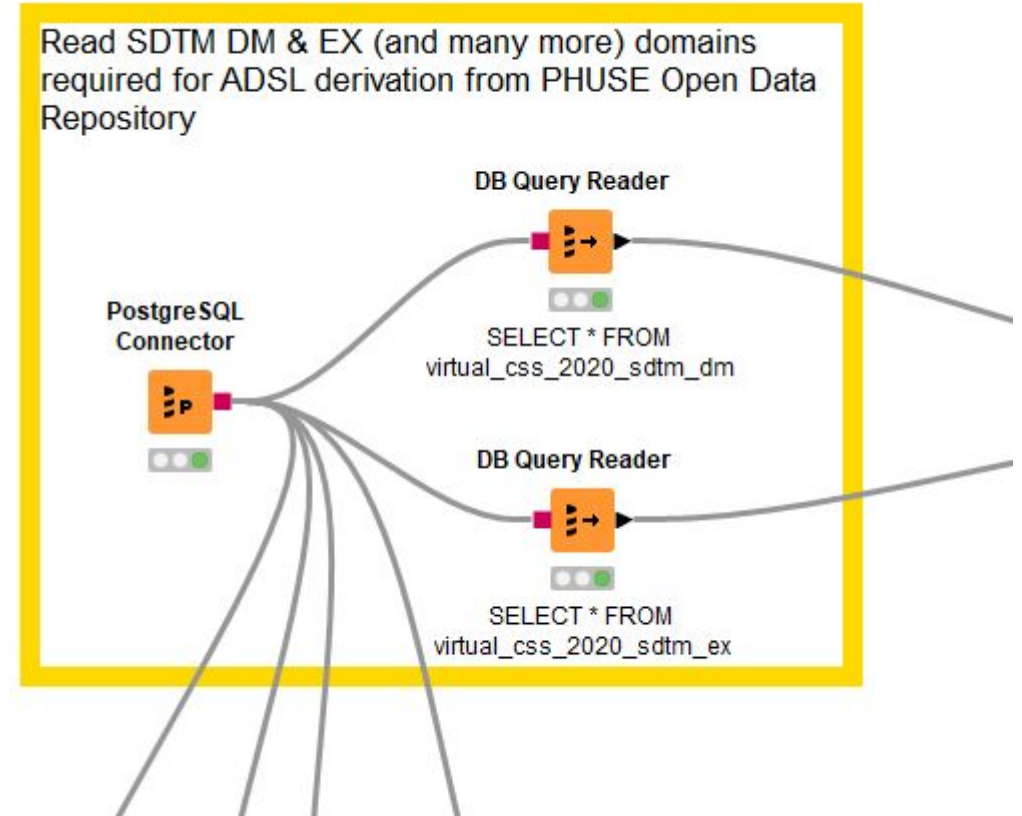


# ADSL – Data Retrieval

<https://github.com/phuse-org/PODR>



- “PODR integrates health-related Open Data across agencies”
- Sample data sets
  - SDTM, ADaM ... and more





# ADSL – Data Retrieval

<https://github.com/phuse-org/PODR>

Dialog - 0:368 - PostgreSQL Connector

File

Output Type Mapping | Flow Variables | Job Manager Selection

Connection Settings | JDBC Parameters | Advanced | Input Type Mapping

Configuration

Database Dialect: PostgreSQL

Driver Name: PostgreSQL [ID: PostgreSQL]

Location

Hostname: podr.phuse.global | Port: 5.432

Database name: nihpo

Authentication

Credentials

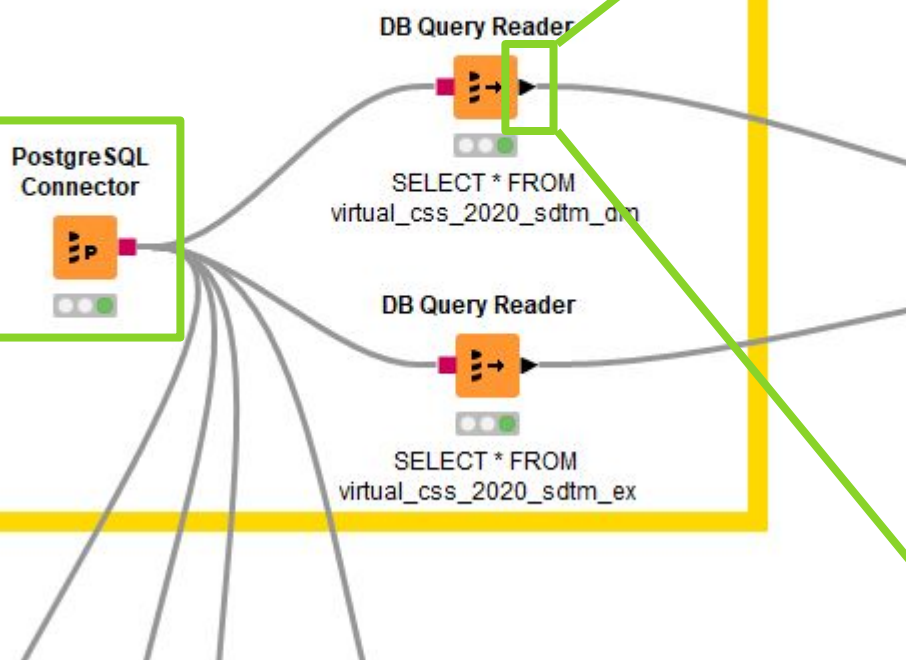
Username & password

Username: phuse\_rfe3ghlh8

Password: ●●●●●●●●

OK | Apply | Cancel | ?

Read SDTM DM & EX (and many more) domains required for ADSL derivation from PHUSE Open Data Repository



KNIME data table - 6:373 - DB Query Reader (virtual\_css\_2020\_sdtm\_dm)

File Edit Hilite Navigation View

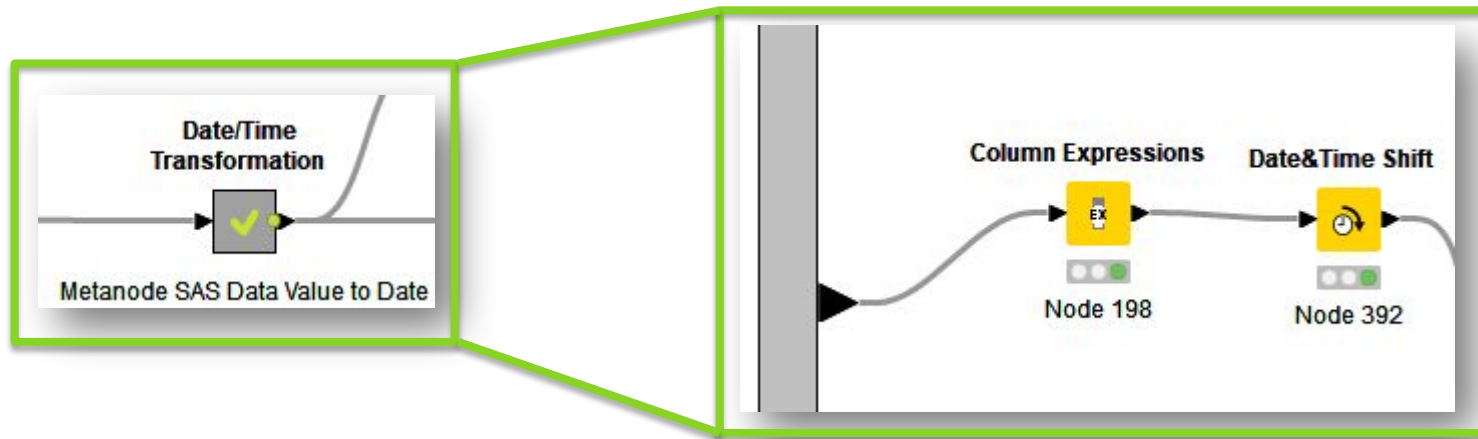
Table "database" - Rows: 306 | Spec - Columns: 25 | Properties | FI

Row ID	S STUDYID	S DOMAIN	S USUBJID
Row0	CDISCPIL01	DM	01-701-1015
Row1	CDISCPIL01	DM	01-701-1023
Row2	CDISCPIL01	DM	01-701-1028
Row3	CDISCPIL01	DM	01-701-1033
Row4	CDISCPIL01	DM	01-701-1034
Row5	CDISCPIL01	DM	01-701-1047
Row6	CDISCPIL01	DM	01-701-1057
Row7	CDISCPIL01	DM	01-701-1097
Row8	CDISCPIL01	DM	01-701-1111
Row9	CDISCPIL01	DM	01-701-1115
Row10	CDISCPIL01	DM	01-701-1118
Row11	CDISCPIL01	DM	01-701-1130
Row12	CDISCPIL01	DM	01-701-1133
Row13	CDISCPIL01	DM	01-701-1145
Row14	CDISCPIL01	DM	01-701-1146
Row15	CDISCPIL01	DM	01-701-1148
Row16	CDISCPIL01	DM	01-701-1153
Row17	CDISCPIL01	DM	01-701-1162
Row18	CDISCPIL01	DM	01-701-1176
Row19	CDISCPIL01	DM	01-701-1180
Row20	CDISCPIL01	DM	01-701-1181
Row21	CDISCPIL01	DM	01-701-1188
Row22	CDISCPIL01	DM	01-701-1192
Row23	CDISCPIL01	DM	01-701-1203

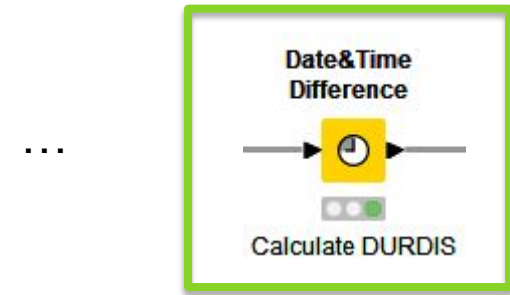


# ADSL – Derivation

## Coping with date formats



Allow date calculations



SAS date representation

19800
19905
19401
19724

Define SAS origin

1960-01-01
1960-01-01
1960-01-01
1960-01-01

Shift to KNIME dates

2014-01-02
2012-08-05
2013-07-19
2014-03-18

Duration from TRTSTD\* (e.g. in days)

44
76
43
55

\*TRTSTD = Treatment Start Date

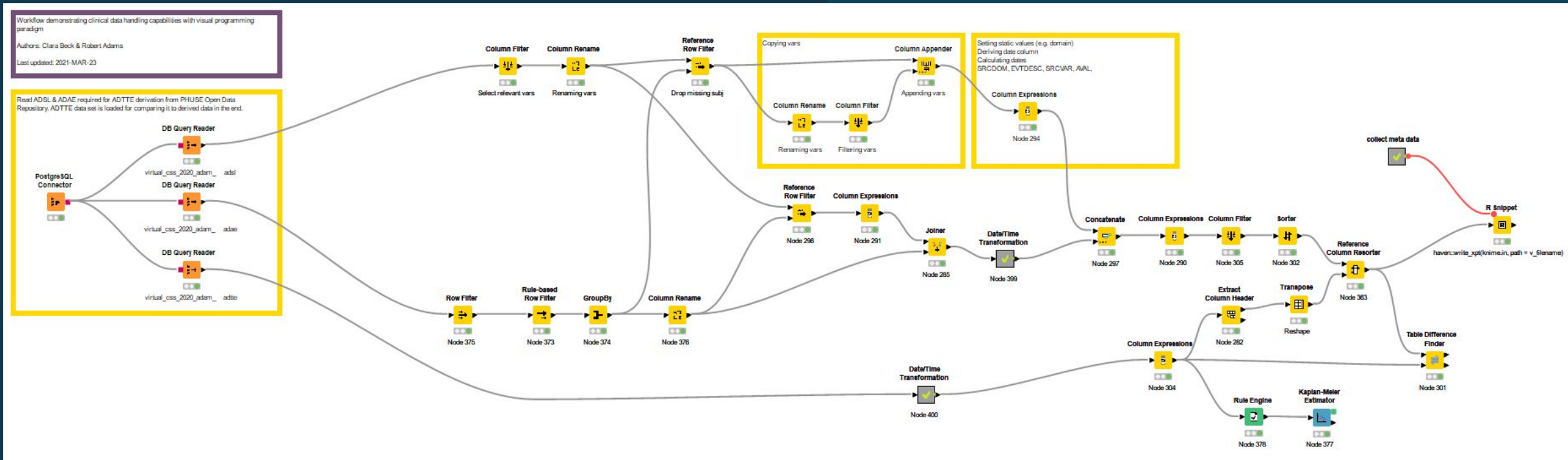


# ADaM Derivation – ADTTE (time to event)

Data retrieval

Derivation

Visualization & Testing & XPT output





# Breakout – Survival Analysis

$$S(t) = P(T \geq t),$$

$$\hat{S}(t) = \prod_{i=1}^{t_i \leq t} \left( \frac{n_i - d_i}{n_i} \right)$$

Crucial **efficacy representation** of clinical trials – „did a drug has an anticipated effect?“

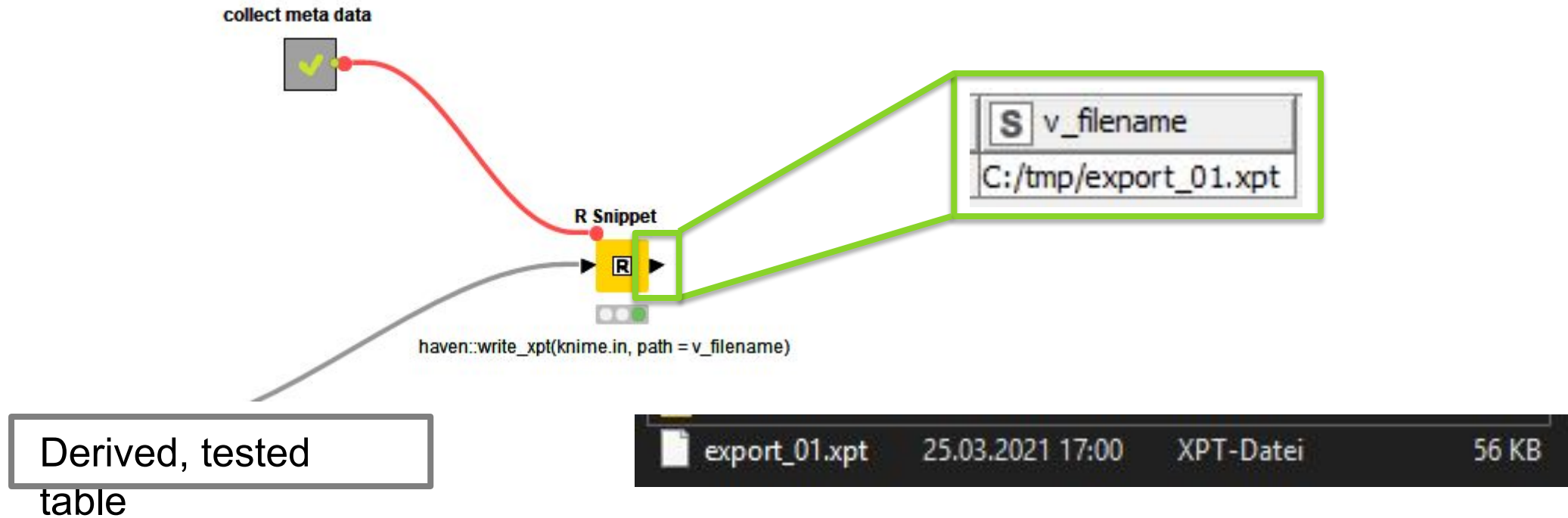
- General: area of **statistics designed for modelling TTE** (time-to-event) data („expected duration of time until event occurs“); applications in sociology, economics, engineering, biology etc.
  - E.g. failure in mechanical systems, **survival of a population past a certain time**
- In oncological clinical study context: analyse the **time to disease remission, progression or death for cohorts of patients** or compare different treatments within a clinical trial
- Events typically subject to **censoring** (missing / incomplete) for variety of reasons
  - I.e. subjects are „**lost to follow-up**“ or **drop out** of a study for reasons independent of survival
  - Influences statistics as power decreases by censoring (uncertainty increases)
- **Survivor function** (probability to experience and event by given time, e.g. survival probability after 24 months) and **harzard rate** (instantaneous risk of a subject experiencing an event at a given time)
  - Wilcoxon test and log-rank test used to calculated differences between groups ( $H_0$ : groups have the same hazard)
  - Usually Kaplan-Meier plots to present survival data are used





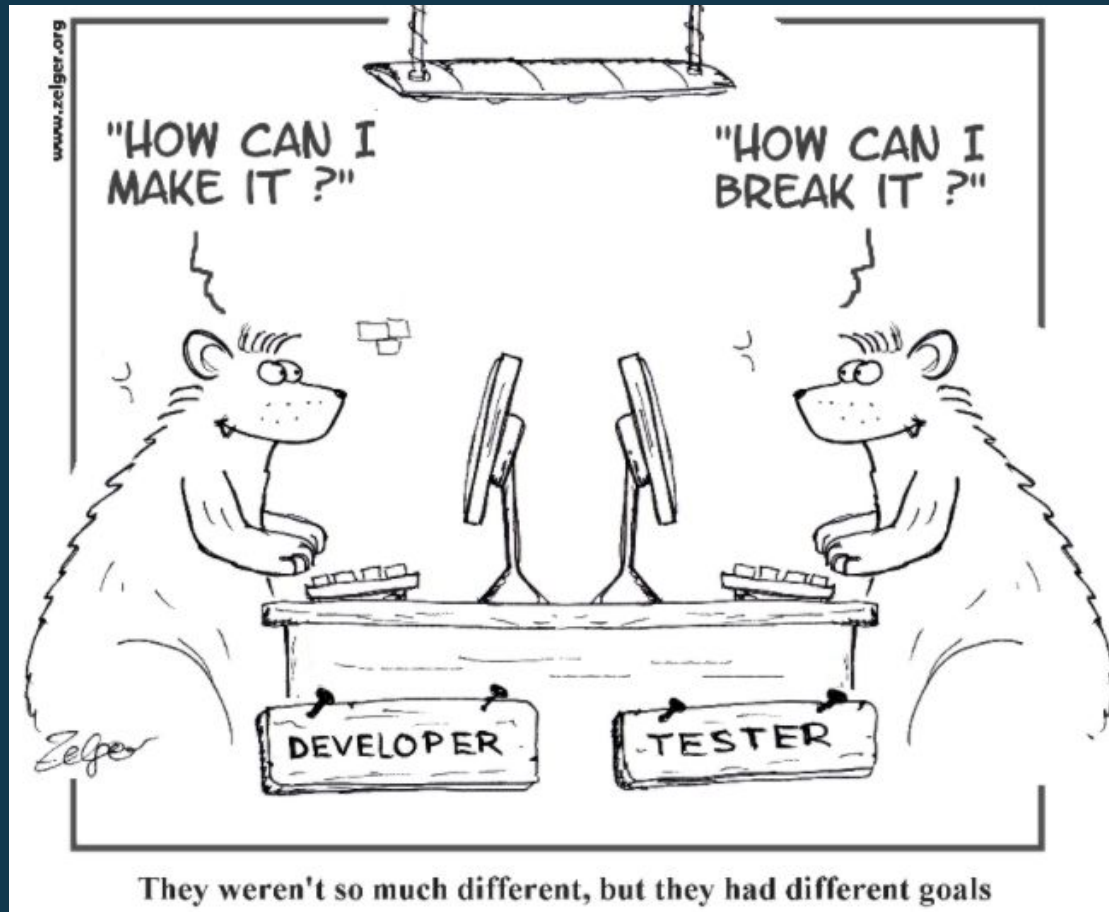
# ADTTE – Derivation

Utilizing R to write XPT output (regulatory authority requirement)



# Core of GxP:

*Validation,  
Testing,  
Logging,  
Reporting*







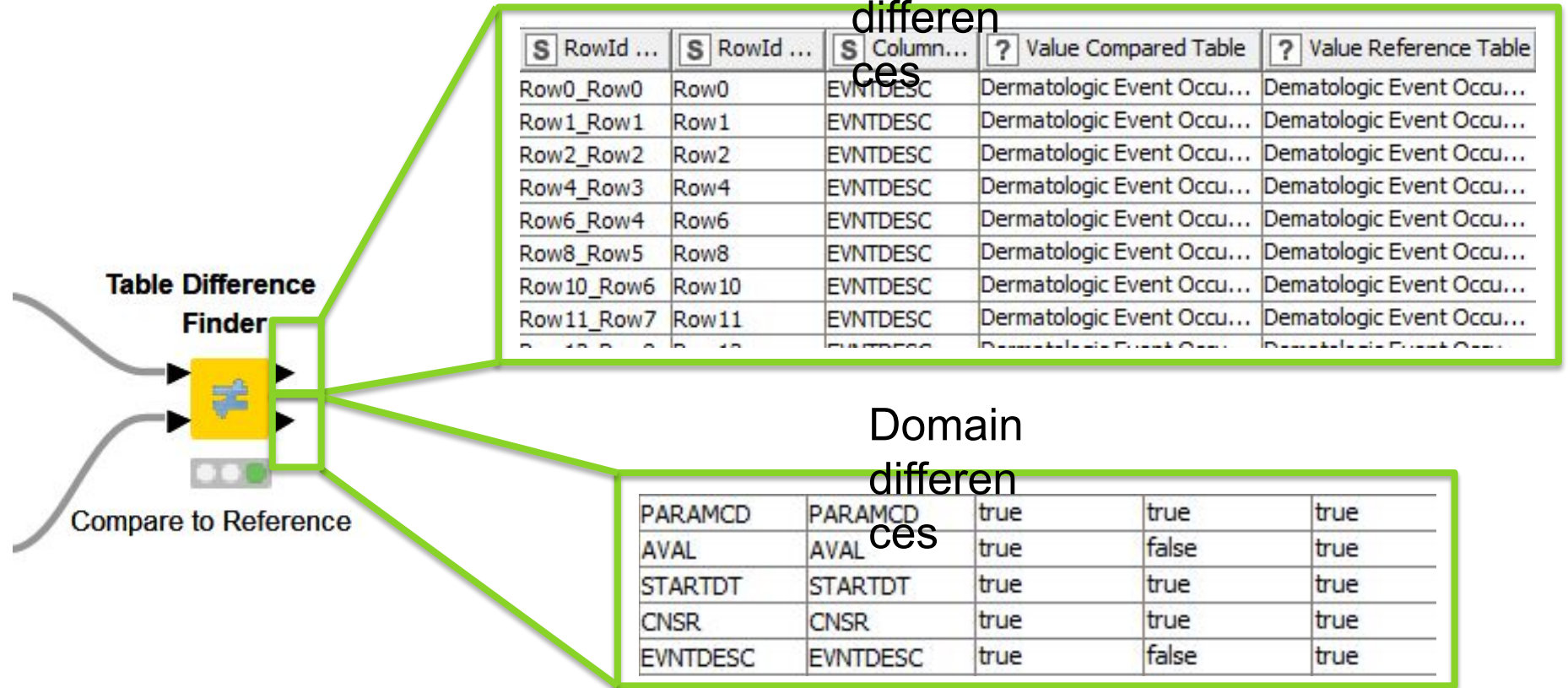
# Double Programming Validation

Dear PODR – no offense ☺ you have a typo!

Derived data table

Gold standard data set;  
PODR ADTTE  
(or SAS derived data)

Domain differences are an additional validation layer



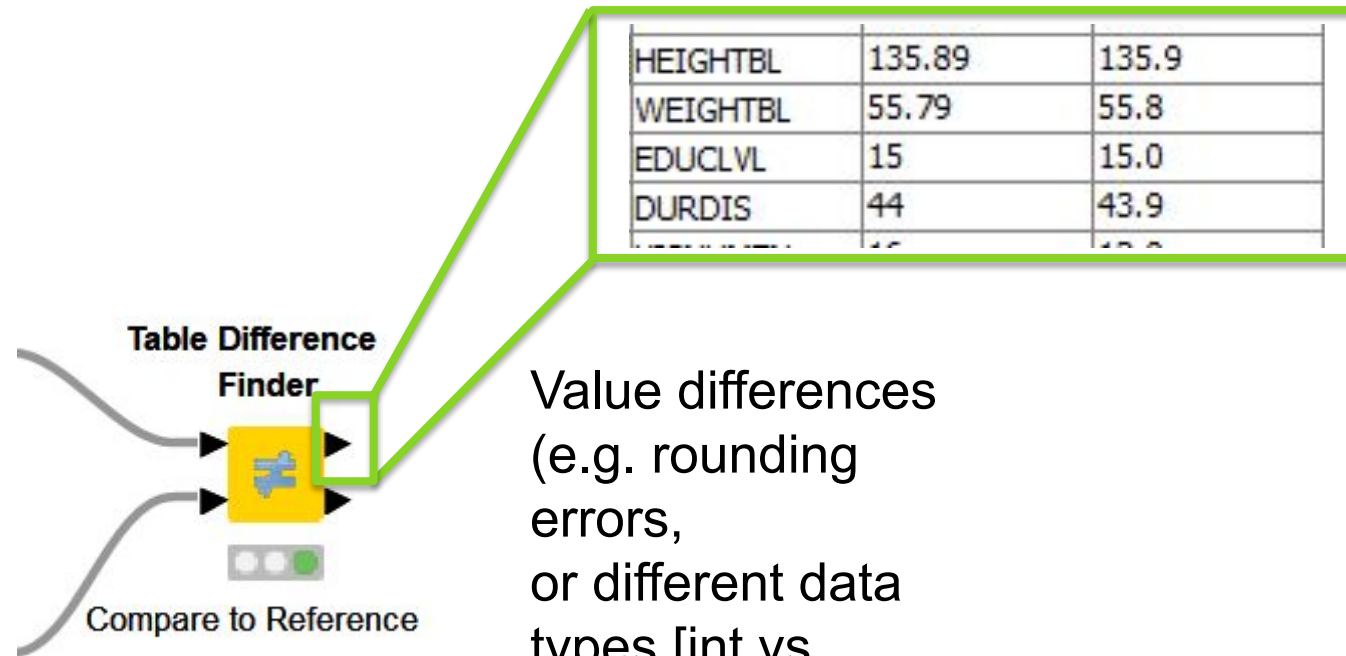
For numeric: min / max (range) differs  
For nominal: different unique values



# Double Programming Validation

Derived data table

Gold standard data set:  
PODR ADTTE  
(or SAS derived data)



Value differences  
(e.g. rounding errors,  
or different data types [int vs. double])

# Testing

<https://www.knime.com/blog/enter-the-era-of-automated-workflow-testing-and-validation>

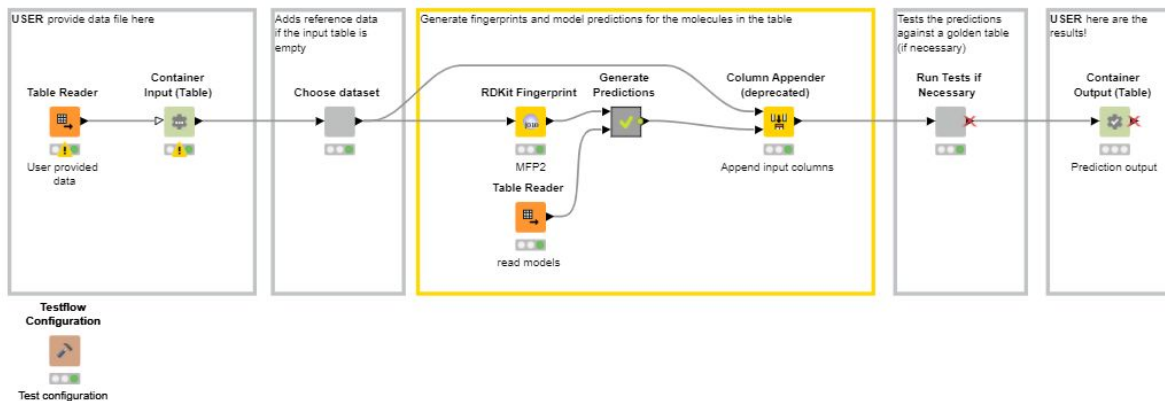
- Testing
  - File Stores and Blobs
    - Create FileStore Column
    - Test FileStore Column
    - Create FileStore Column in LoopEnd
    - Test FileStore Port Object Loop End
    - Test FileStore Port Object to Table
  - Create Test Blobs
  - Verify Test Blobs
  - Block Programmatically
  - Count Execution Programmatically
  - Credentials Validate Test
  - Database Connection Closer
  - Disturber Node
  - Fail in execution
  - File Difference Checker
  - File Difference Checker (Labs)
  - Image Difference Checker
  - Logger Option
  - Model Content Difference Checker
  - PMML Difference Checker
  - Table Difference Checker
  - Test Data Generator
  - Testflow Configuration

## Validating KNIME Workflows

reproducibility validation testing XGBoost RDKit

Last edited: 26 Feb 2021

This workflow demonstrates a technique to deploy a model prediction workflow as a web service and to validate that it is doing what it's supposed to do.



All relevant testing scenarios can be covered



# Logging

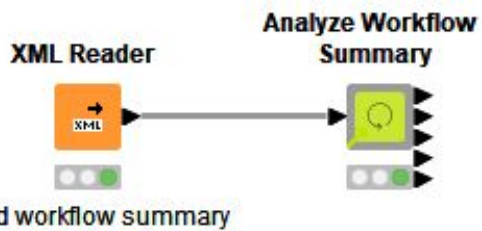
Export executed workflow summary



Verbose XML document traces all data points / transformations throughout a workflow

```
rs > gbeck > Downloads > adslsummary.xml > WorkflowSummary > environment > installation > plugins > plugin
<WorkflowSummary version="1.0.0" summaryCreationDateTime="2021-03-22 14:45:37 +0100"><environment knimeVersion="4.3.2.v202103051236" os="Windows 10"><installation><plugins><plugin
name="org.eclipse.osgi" version="3.15.200.v20200214-1600"/><plugin name="org.eclipse.osgi.compatibility.state" version="1.1.700.v20200207-2156"/><plugin name="javax.transaction"
version="1.1.1.v201105210645"/><plugin name="org.eclipse.equinox.simpleconfigurator" version="1.3.500.v20200211-1505"/><plugin name="cc.mallet" version="2.0.8.v20180913-kgme"/
><plugin name="com.amazonaws.aws-java-sdk-core" version="1.11.602"/><plugin name="com.amazonaws.aws-java-sdk-s3" version="1.11.602"/><plugin name="com.amazonaws.aws-java-sdk-sts"
version="1.11.602"/><plugin name="com.epam.parso" version="2.0.2"/><plugin name="com.fasterxml.jackson.core.jackson-annotations" version="2.11.0"/><plugin name="com.fasterxml.
jackson.core.jackson-core" version="2.11.0"/><plugin name="com.fasterxml.jackson.core.jackson-databind" version="2.11.0"/><plugin name="com.fasterxml.jackson.dataformat.
jackson-dataformat-cbor" version="2.11.0"/><plugin name="com.fasterxml.jackson.dataformat.jackson-dataformat-xml" version="2.11.0"/><plugin name="com.fasterxml.jackson.datatype.
jackson-datatype-jdk8" version="2.11.0"/></plugins></installation></environment></WorkflowSummary>
```

Use third party tools or read back into KNIME and use the provided Analyzer component



Plugin Name	Version
org.knime.workbench.workflow...	4.3.0.v202011191609
org.knime.workflow.migration	4.3.0.v202011191336
org.knime.xml	4.3.0.v202011212018
org.mortbay.jetty.server	6.1.26
org.mortbay.jetty.util	6.1.26
org.mozilla.javascript	1.7.5.v201504281450
org.objectweb.asm	5.0.4
org.openqa.selenium	3.12.0.v202010300...
org.reactivestreams.reactive-...	1.0.3
org.sat4j.core	2.3.5.v201308161310

Installed plugins

Node Name	Type	Count
Column Expressions	Manipulator	13
Column Filter	Manipulator	2
Column Rename	Manipulator	10
DB Query Reader	Source	9
Date&Time Differ...	Manipulator	2
Date/Time Shift (!...	Manipulator	4
Extract Column H...	Manipulator	1
Extract Context P...	Source	1
Extract System Pr...	Source	1
GroupBy	Manipulator	4
Java Snippet (sim...	Manipulator	1

Used nodes

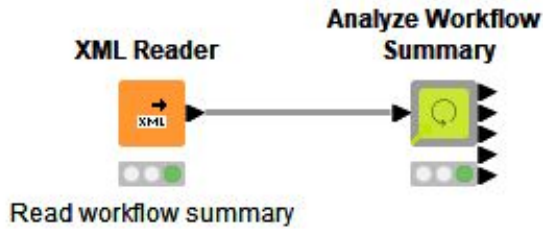
S Knime Version	S OS	S WF Author	S Creatio...	S Last Edi...	S Last Up...	S WF Description	S summaryCreationDate...	S WF Title	[...] WF An...
4.3.2.v202103051236	Windows 10	Robert Adams & Clara Beck	2021-03-02	?	?	PHUSE_Workflow_ADSL_PODR Demonstrates ADSL derivation from PODR data sets for PHUSE US Connect 2021	2021-03-23 11:44:46 +0100	PHUSE_Workflow_ADSL_PODR	[]

## Environment & Metadata



# Logging

## Export executed workflow summary



Node details:

- For each node input, state, configuration settings, variables

• Data at input

Node details - 5:0 - Analyze Workflow Summary

File Edit Hilite Navigation View

Table "default" - Rows: 81692 Spec - Columns: 27 Properties Flow Variables

Row ID	B isComp...	B Deprec...	S parent...	B isMetan...	S OutputPort	S succee...	[...] precedin...	[...] inputPort	S key	S value	S out_po...	S out_po...	S col_index	S col_name	S col_type	[M] column...	[M] column...
Row48	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table	7	VSORRES	Number (do...	<?xml v...	<?xml v...
Row49	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table	8	VSORRESU	String	<?xml v...	<?xml v...
Row50	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table	9	VSSTRESC	Number (do...	<?xml v...	<?xml v...
Row51	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table	10	VSSTRESN	Number (do...	<?xml v...	<?xml v...
Row52	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row53	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row54	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row55	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row56	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row57	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row58	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row59	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row60	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row61	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row62	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row63	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					
Row64	false	false	Top Level	false	output port 1 (table)	?	[PostgreSQL Connector:368: output port 1]	[input port 1 (PortTy...	sql_statement	SELECT * F...	1	table					

KNIME Console

```

WARN Math Formula 5:2 No configuration available
WARN PostgreSQL Connector 7:368 DB Connection no longer available. Go to advanced settings to
WARN Column Expressions 7:8 An error occurred in script 1:
The provided date must not be null (at line 0, function: 'date')
ERROR Date/Time Shift (legacy) 0:306:199 Execute failed: org.knime.core.data.def.DoubleCell cannot
ERROR Date/Time Shift (legacy) 0:307:199 Execute failed: org.knime.core.data.def.DoubleCell cannot
WARN PostgreSQL Connector 8:368 DB Connection no longer available. Go to advanced settings to
WARN Column Expressions 8:8 An error occurred in script 1:
The provided date must not be null (at line 0, function: 'date')
WARN Column Expressions 8:8 An error occurred in script 1:

```



## Conclusion & Take-aways



- Powerful (feature complete) alternative for working with (clinical) data
- Clinical data derivation possible
  - Alternative approach for double programming / validation of SAS derivations
  - Verbose logging & validation capabilities
- Visual programming especially for pipeline optimization, automation & educational purposes with great potential



- Highly regulated environment and current standards
- Industry adoption
- Existing pipelines, SOPs and study continuity



# Acknowledgements



**Seema Nair**

Bayer – Onc SBU – Statistical Programming



**Anupama Sheoran**

Bayer – Onc SBU – Statistical Programming



**José C. Lacal**

PHUSE – PODR



**Cornelia Fulgenzi**

Bayer – Onc SBU – Statistical Programming



# Contact Information

**Robert Adams**

Lead Computational Scientist

Bayer – Oncology Digitalization & Computational Science



**Clara Beck**

Senior Computational Scientist



[robert.adams@bayer.com](mailto:robert.adams@bayer.com)





## KNIME Data Talks

Clinical Analysis Dataset Derivation using Visual Programming with KNIME

**Main Author:** Robert Adams  
**Co-Author:** Clara Beck



**Thank you for the attention**

