

RE04: Enhancing the Validation and Simulation of Real-World Multi-Site Data with R

PHUSE EU Connect 2025

Nikolay Trankov, Elena Chaparova, Alexandrina Lambova,
Stavros Oikonomou, Monika Petkova, Louise Raiteri



Agenda

- + Challenges of working with Real-World Data
- + Traditional workflow
- + The Common Data Specification (CDS)
- + R package
- + Validation
- + Simulation
- + Automated CDS reading

Challenges of working with Real-World Data (RWD)



Data for RW studies is frequently sourced across multiple hospitals situated in different countries

- Due to differences between practices and regions, data comes in a variety of formats
- There may be differences in data availability between different data sources



Timelines for RWE studies are subject to delays

- Site set up and time to data receipt can be longer than expected
- Site communication is often a lengthy and iterative process



RWD is imperfect and needs to be validated before it can be analysed

- Validation done by hand is time consuming
- Complex data validation rules can be difficult to execute

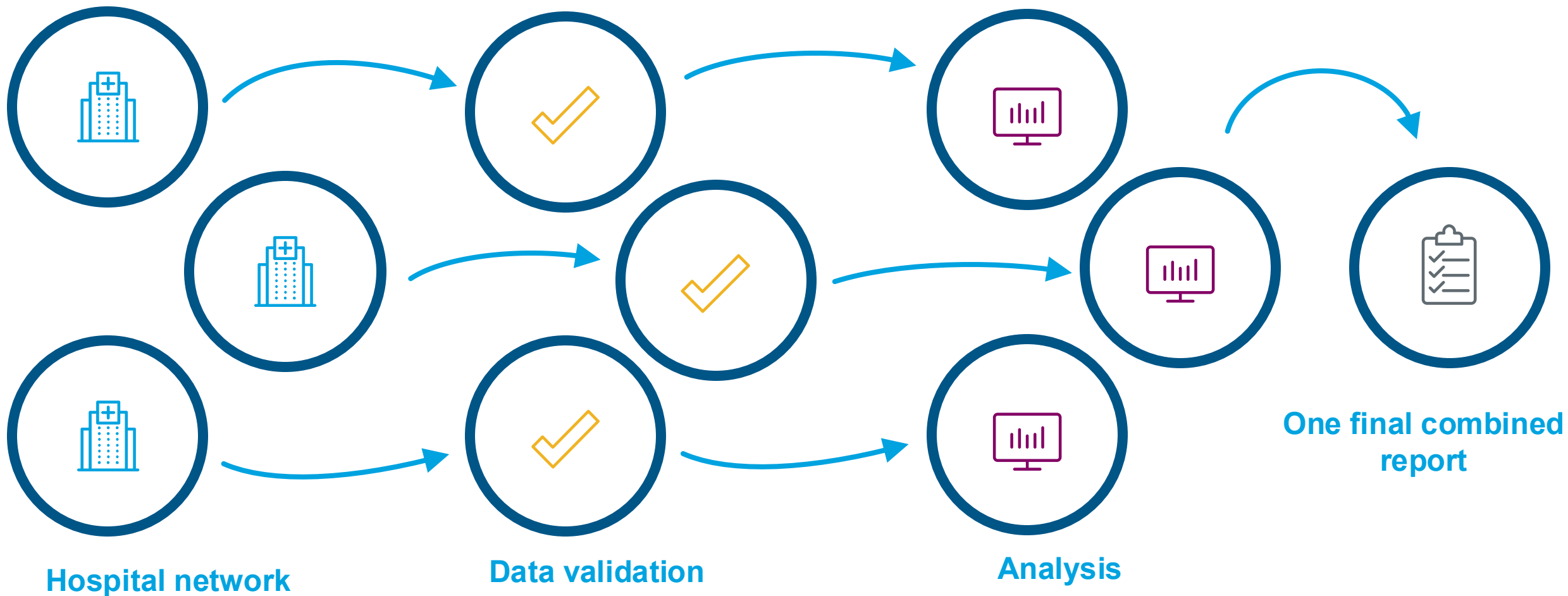


Analysing data from multiple data sources

- Analysis done separately for each data source is inefficient and prone to mistakes

Traditional workflow

Data from multiple sites goes through separate validation and separate analyses, before being combined to a single report



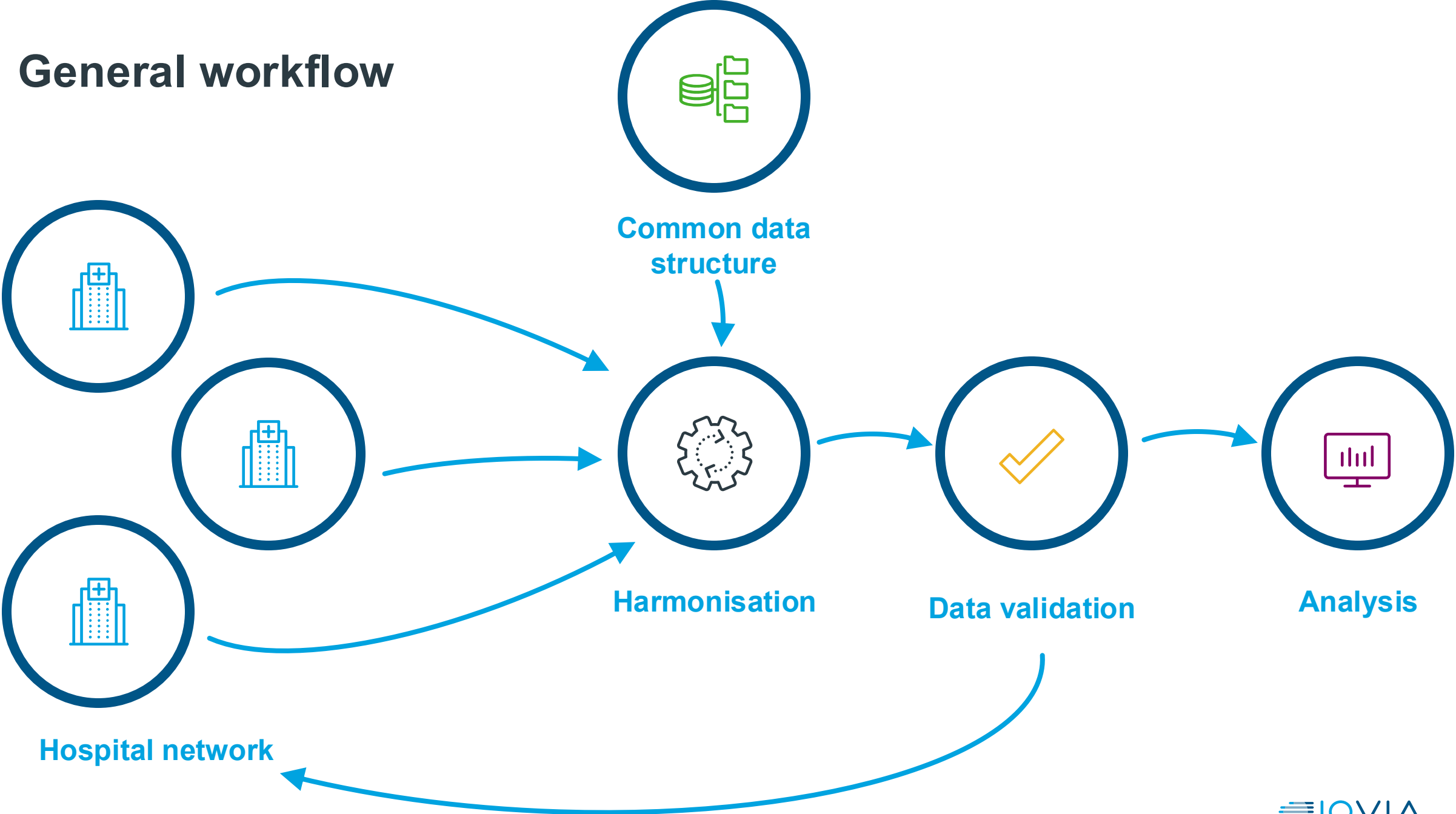
The Common Data Specification (CDS)

The CDS is an excel based document, containing all variables to be extracted for a study

- The CDS is used to harmonise data across countries and sites
- It contains various information related to variable type, timepoint or time window, structure and format
- Contains logic checks to ensure data quality

4	The Patient table reports all demographic and				
5	Combination of PTID & SITE				
6	Data Variable	CDM Variable Code	Description	Type	Valid Values and Mapping
7	Patient ID	PTID	Study-specific patient ID.	num	Unique
8	Site ID	SITE	Unique centre identifier	char	A = Site 1 B = Site 2 C = Site 3 D = Site 4
9	Review of manual abstraction	REVIEW	Method of review for manual abstraction. Select Four-eyes if person A types up data and person B checks the manually typed data against the source records Select four-eyes independent if person A types up data, person B types up same data and person A & B both check differences Select other if differs from above.	cat	Four-eyes Four-eyes independent Other
10	Date of birth	BIRTHDAT	Patient's date of birth	dat	yyyymm
11	Initial diagnosis date	DIAGDAT	Initial date of diagnosis of functional or non-functional G1, G2 or well-differentiated G3 GEP-NETs	dat	yyyymmdd

General workflow



R package for data analysis and reporting

Which has additionally been extended to include data validation and simulation functionality

The DataStruct object is an abstraction for a data set and defines the metadata for each variable

```
data_structure <- DataStruct(  
  variables = list(  
    NumVariable(  
      name = "age",  
      label = "Age at diagnosis"  
    ),  
    CatVariable(  
      name = "gender",  
      label = "Sex",  
      categories = list(list(1, c("Male", "Female")))  
    ),  
    DateVariable(  
      name = "diag_date",  
      label = "Date of diagnosis"  
    )  
  )  
)
```

NumVariable

Represents continuous data

CatVariable

Represents data with several levels and with or without hierarchical structure

DateVariable

Represents dates data

Validation functionality

A set of functions which allow the user to define multiple validation rules for a variable and to create a report based on the rules

```
data_structure <- DataStruct(  
  variables = list(  
    NumVariable(  
      name = "age",  
      label = "Age at diagnosis",  
      validateFunc = list(  
        vals_between_values(  
          left = 18, right = 105, na_pass_validation = FALSE)  
        ),  
      CatVariable(  
        name = "gender",  
        label = "Sex",  
        categories = list(list(1, c("Male", "Female"))),  
        validateFunc = list(  
          vals_not_na()  
        ),  
      DateVariable(  
        name = "diag_date",  
        label = "Date of diagnosis",  
        validateFunc = list(  
          vals_between_cols(  
            left_col = "study_start_date",  
            right_col = "study_end_date"),  
          vals_less_than_or_equal_col(  
            col = "death_date",  
            na_pass_validation = TRUE))  
      )  
    )  
  )  
)
```

NumVariable	CatVariable	DateVariable
col_exists	col_exists	col_exists
integer_vals	character_vals	date_vals
numeric_vals	logical_vals	posix_vals
unique_vals	factor_vals	vals_in_set
vals_in_set	unique_vals	vals_less_than_value
vals_not_in_set	vals_in_set	vals_less_than_col
vals_less_than_value	vals_not_in_set	vals_less_than_or_equal_value
vals_less_than_col	vals_equal_value	vals_less_than_or_equal_col
vals_less_than_or_equal_value	vals_match_regex	vals_equal_value
vals_less_than_or_equal_col	vals_na	vals_equal_col
vals_equal_value	vals_not_na	vals_greater_than_value
vals_equal_col	vals_expr	vals_greater_than_col
vals_greater_than_value		vals_greater_than_or_equal_value
vals_greater_than_col		vals_greater_than_or_equal_col
vals_greater_than_or_equal_value		vals_between_values
vals_greater_than_or_equal_col		vals_between_cols
vals_between_values		vals_increasing
vals_between_cols		vals_decreasing
vals_increasing		vals_na
vals_decreasing		vals_not_na
vals_na		vals_expr
vals_not_na		
vals_expr		

The validate report – an HTML-based report using the pointblank R package

Pointblank Validation

[2025-10-07|12:31:41]

DATA FRAME

data

WARN1

STOP1

NOTIFY-

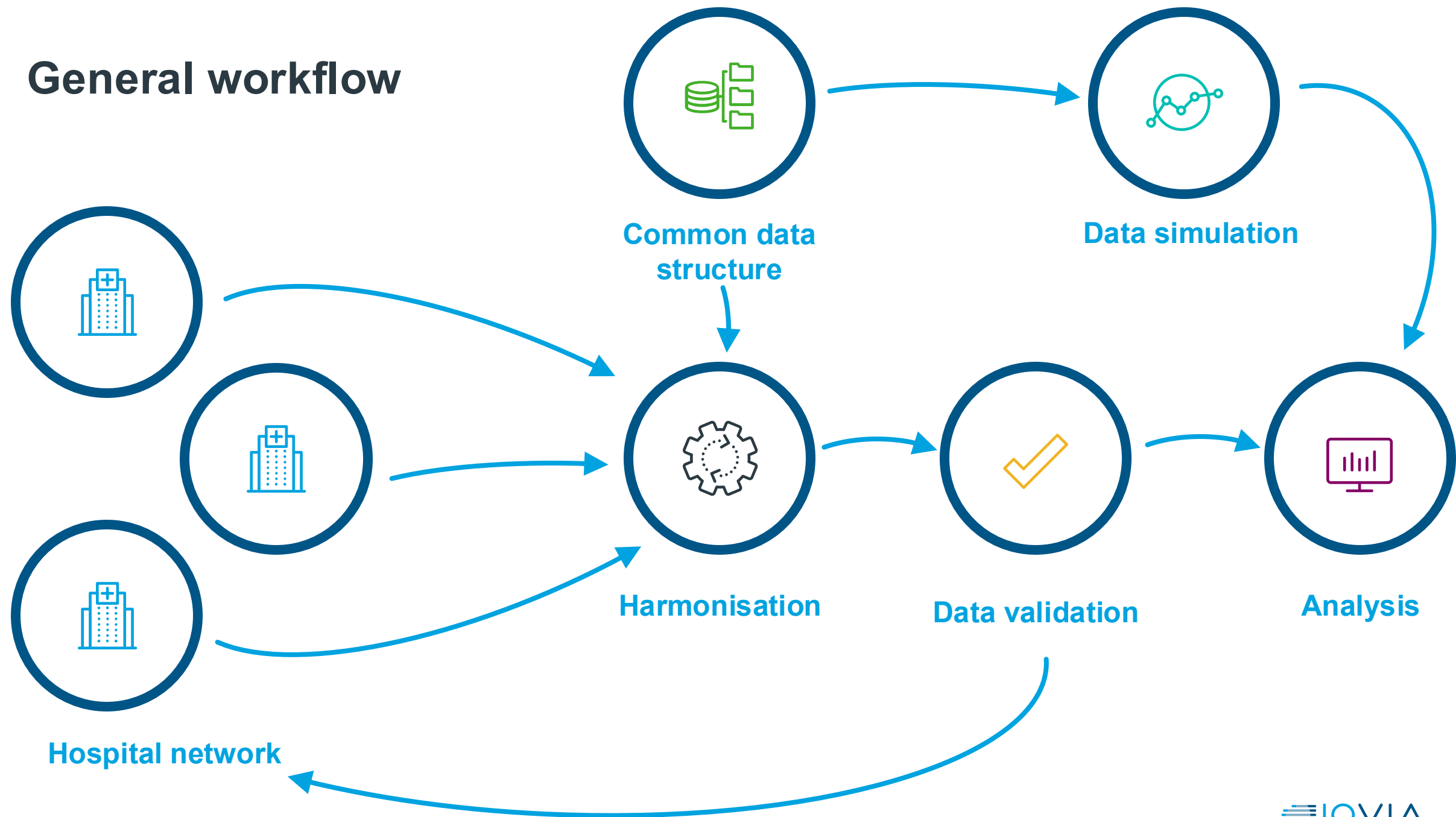
STEP		COLUMNS	VALUES	TBL	EVAL	UNITS	PASS	FAIL	W	S	N	EXT
1	Values between values()	age	[18, 105)	→	✓	100	80 0.80	20 0.20	●	●	—	CSV
2	Integer values()	age	—	→	✓	1	1 1.00	0 0.00	○	○	—	—
3	Column exist()	gender	—	→	✓	1	1 1.00	0 0.00	○	○	—	—
4	Not NA()	gender	—	→	✓	100	90 0.90	10 0.10	●	●	—	CSV
5	Values between columns()	diag_date	[study_start_...	→	✓	100	7 0.07	93 0.93	●	●	—	CSV
6	Less than or equal to values in column death_date()	diag_date	~death_date	→	✓	100	61 0.61	39 0.39	●	●	—	CSV
7	Not NA()	diag_date	—	→	✓	100	90 0.90	10 0.10	●	●	—	CSV

2025-10-07 12:31:41 EEST

2.2 s

2025-10-07 12:31:43 EEST

General workflow



Simulation functionality

A set of functions which allows the user to create synthetic data for testing and analytical purposes

```
data_structure <- DataStruct(  
  variables = list(  
    NumVariable(  
      name = "age",  
      label = "Age at diagnosis",  
      simulateFunc = generate_range(  
        min = 18, max = 100, missing = 0.2)  
    ),  
    CatVariable(  
      name = "gender",  
      label = "Sex",  
      categories = list(list(1, c("Male", "Female"))),  
      simulateFunc = generate_nested_sample(  
        categories = list(  
          list(1, "Male"),  
          list(1, "Female"))  
    ),  
    DateVariable(  
      name = "diag_date",  
      label = "Date of diagnosis",  
      simulateFunc = generate_dates(  
        from = as.Date("1970-01-01"),  
        to = as.Date("2023-01-01"),  
        by = "day",  
        missing = 0.1)  
    )  
  )  
)
```

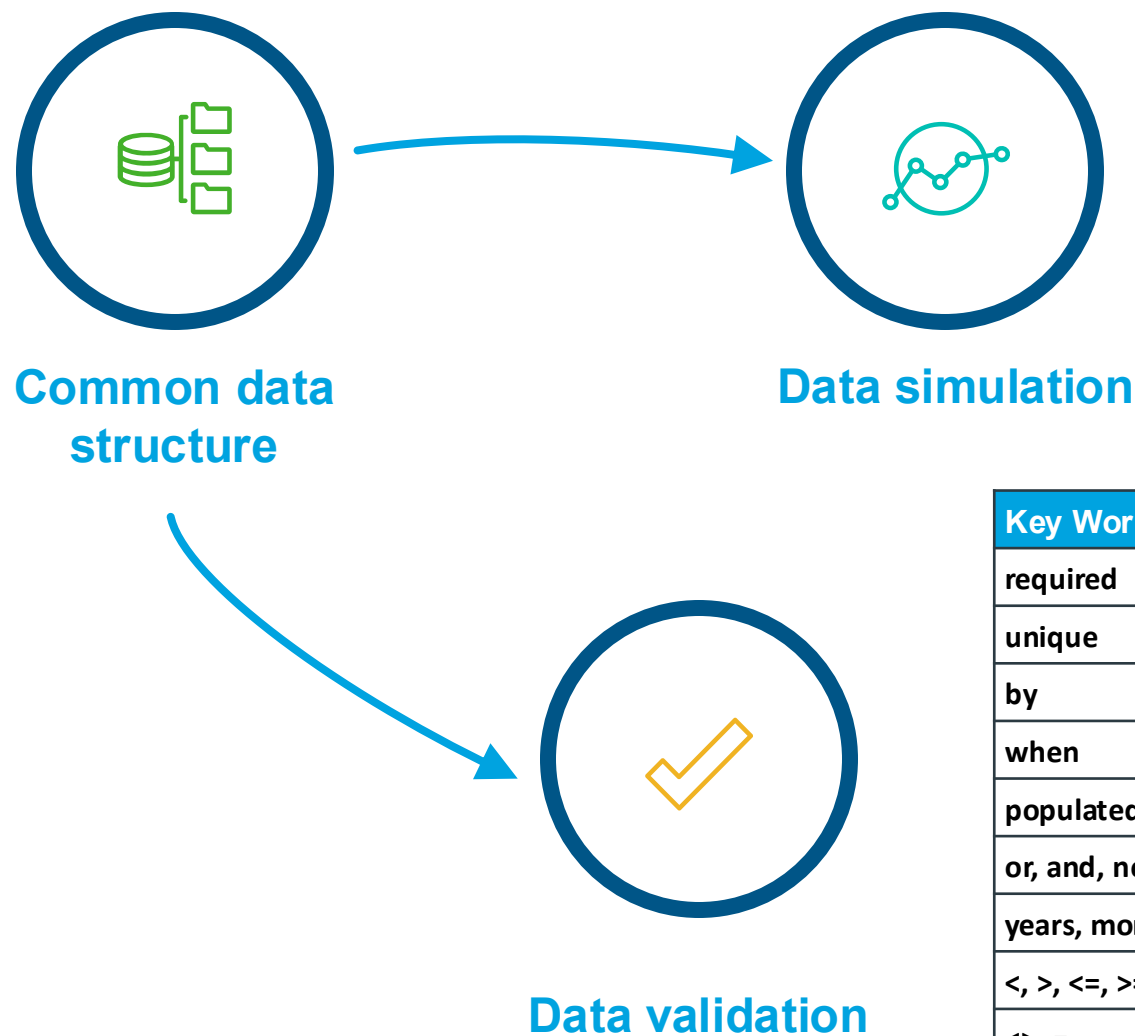
To simulate a dataset, call the simulate function:

```
simulated_data <- simulate(data_structure, n = 100)  
head(simulated_data)
```

```
> head(simulated_data)
```

	age	gender	diag_date
	<int>	<char>	<Date>
1:	NA	Male	2008-01-01
2:	45	Female	2016-01-01
3:	58	<NA>	1980-01-01
4:	92	Female	1988-01-01
5:	52	Female	2017-01-01
6:	50	Male	1975-01-01

Automating the validation and simulation functionality

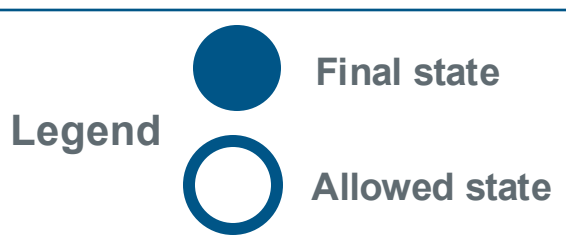
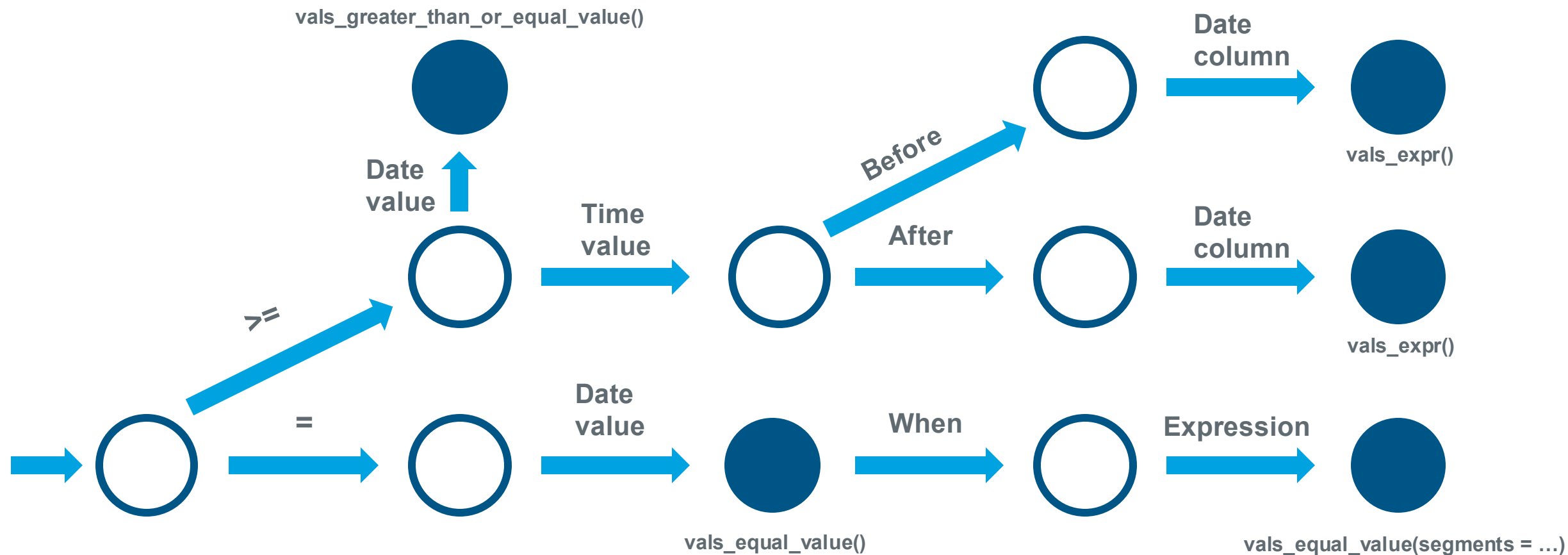


- Standardised CDS format
- Automated reading of the CDS in R
- Automated generation of the validation functions based on the CDS logic checks
- Automated generation of the simulation functions based on the variable types and values
- Dictionary of reserved key words for parsing

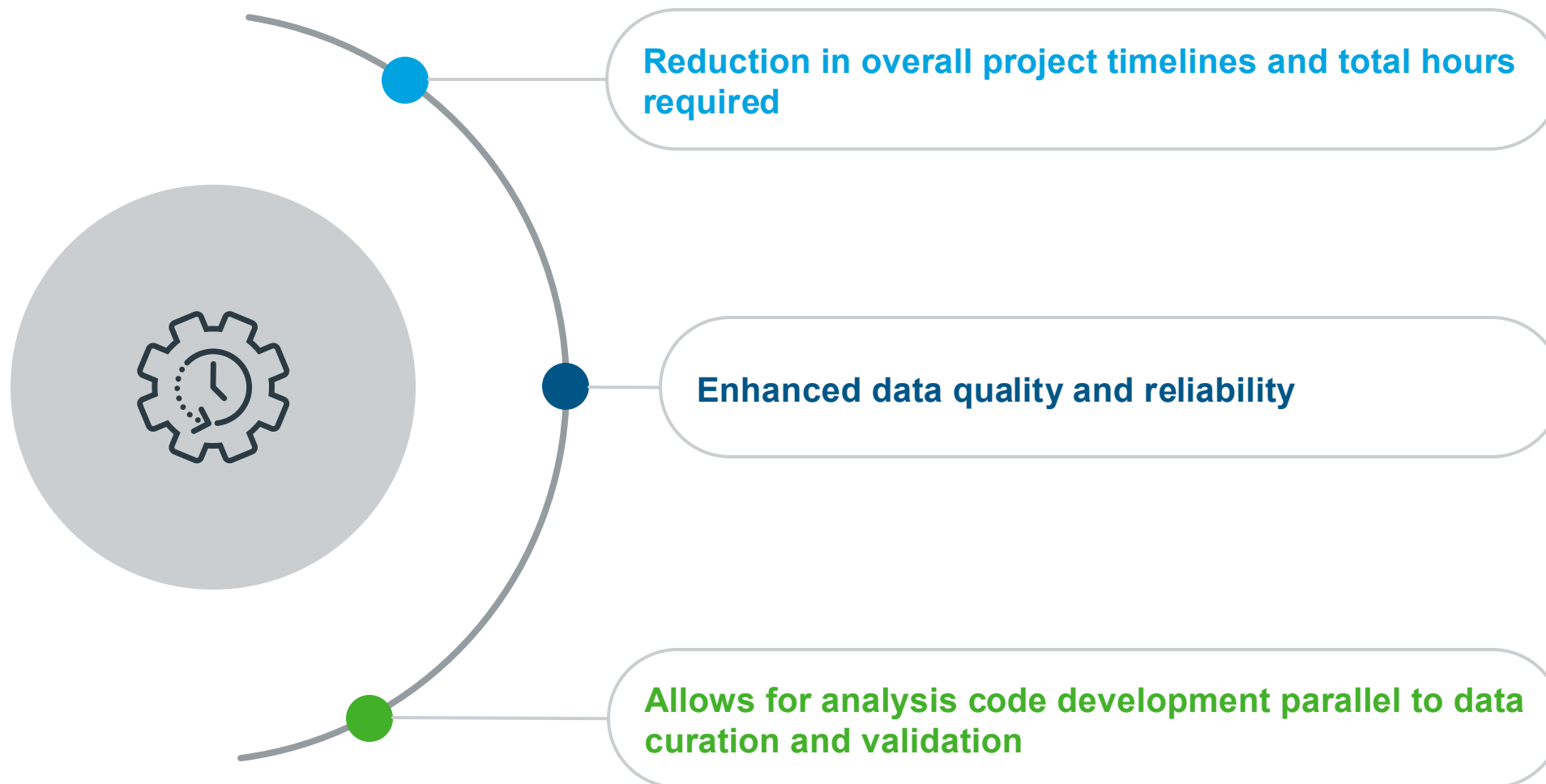
Key Word	Explanation
required	Required variables can't have missing values
unique	Values in the variable must be unique
by	The operation will be performed by group based on another column(s)
when	The operation will be performed on the condition that another expression holds true
populated	Variable can't have missing values, used in conjunction with when
or, and, not	Standard logical operators
years, months, days	Defines a unit of time
<, >, <=, >=	Standard operators for date or numeric comparisons
<>, =	Standard operators for all variable types
in ("val1", "val2")	Value is in set

CDS parsing of validation rules

Finite state machine implemented as a graph in R



Efficiency gains





Questions?



**Thank you for the attention! To
learn more about us, please
contact:**

emeaiqvirdev@iqvia.com



Disclaimer

The analyses, their interpretation, and related information contained herein are made and provided subject to the assumptions, methodologies, caveats, and variables described in this report and are based on third party sources and data reasonably believed to be reliable. No warranty is made as to the completeness or accuracy of such third party sources or data.

As with any attempt to estimate future events, the forecasts, projections, conclusions, and other information included herein are subject to certain risks and uncertainties, and are not to be considered guarantees of any particular outcome.

All reproduction rights, quotations, broadcasting, publications reserved. The contents of this presentation are confidential, and no part of this presentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without express written consent of IQVIA.

Copyright © 2025 IQVIA. All rights reserved. IQVIA® is a registered trademark of IQVIA Inc. in the United States and various other countries.