Automating Data Validation on SQL Server Using R and the Machine Learning package

Paula Hartung

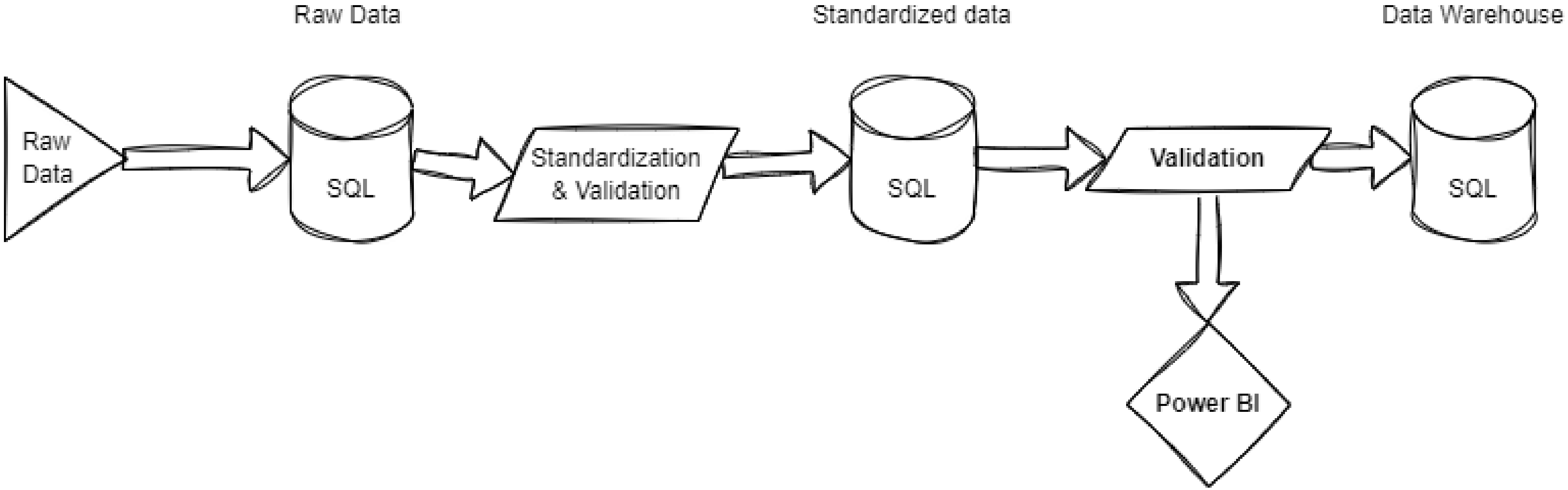
28.11.2024

Statistics Iceland

Informed society

Automated data validation within dataflow on SQL Server

Fully automated data flow in SQL Stored Procedures

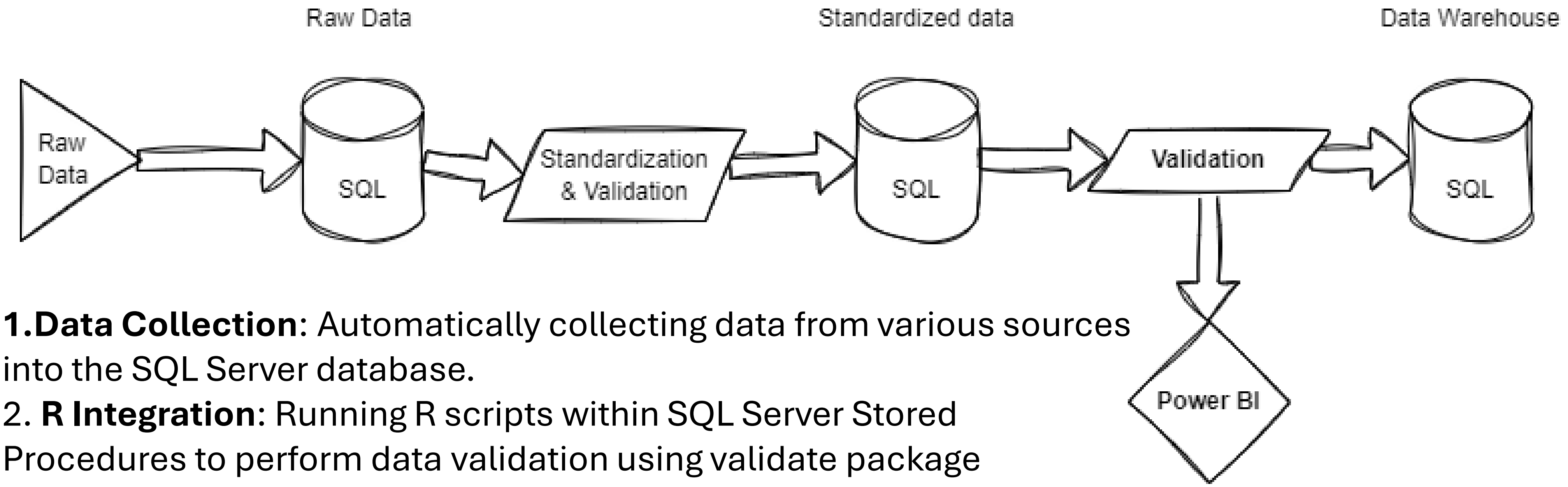


Statistics Iceland

Informed society

Automated data validation within dataflow on SQL Server

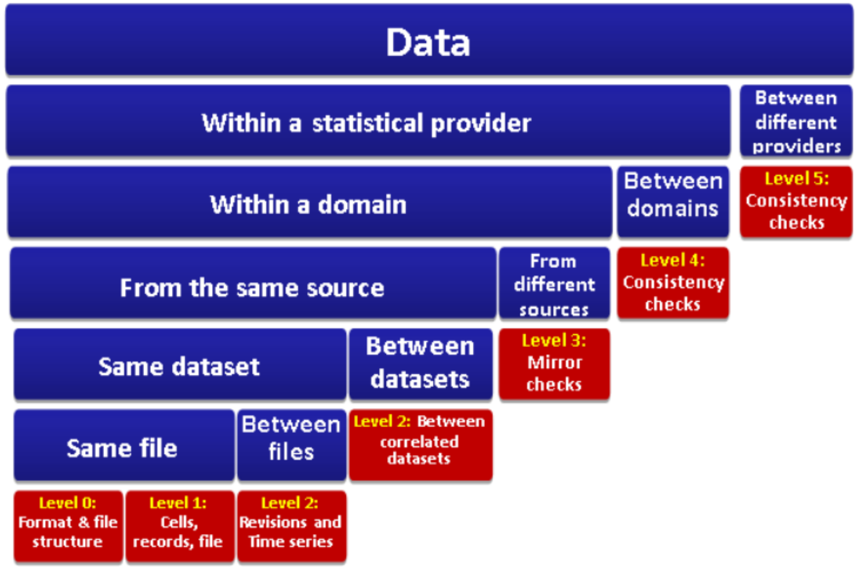
Fully automated data flow in SQL Stored Procedures



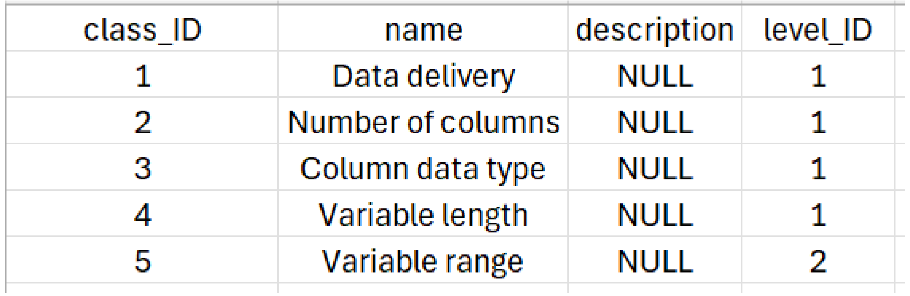
1.Data Collection: Automatically collecting data from various sources into the SQL Server database.
2. **R Integration**: Running R scripts within SQL Server Stored Procedures to perform data validation using validate package
3. **Scheduling**: Setting up SQL Server Agent jobs to automate the execution of Stored Procedures at defined intervals.

Developement of the process:
1 Validation rules in [rules]

- R code directly saved in SQL Server table for every rule
- Validation rules according to validation classes, e.g. Cellular, within data set, against other files, data sets,….
- In collaboration with the (end)users

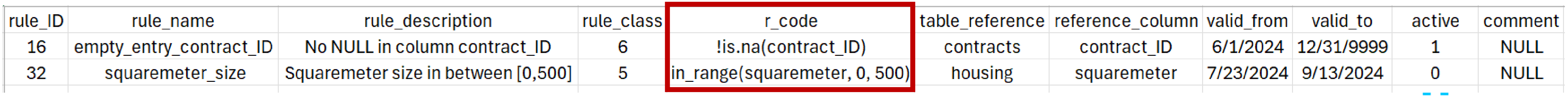

Data
Within a statistical provider
Between different providers
Within a domain
Between domains
Level 5: Consistency checks
From the same source
From different sources
Level 4: Consistency checks
Same dataset
Between datasets
Level 3: Mirror checks
Same file
Between files
Level 2: Between correlated datasets
Level 0: Format & file structure
Level 1: Cells, records, file
Level 2: Revisions and Time series

[validation].[classes]



class_ID	name	description	level_ID
1	Data delivery	NULL	1
2	Number of columns	NULL	1
3	Column data type	NULL	1
4	Variable length	NULL	1
5	Variable range	NULL	2

[validation].[rules]

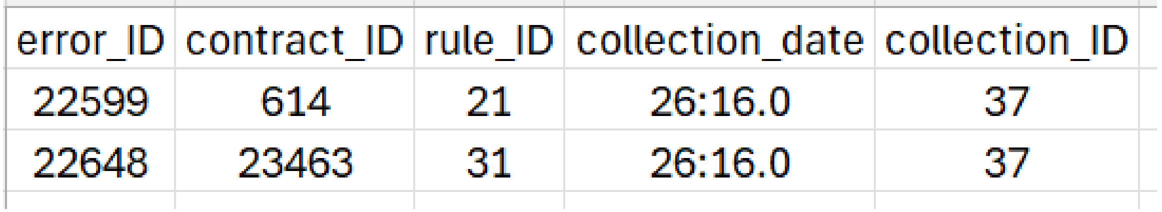


rule_ID	rule_name	rule_description	rule_class	r_code	table_reference	reference_column	valid_from	valid_to	active	comment
16	empty_entry_contract_ID	No NULL in column contract_ID	6	!is.na(contract_ID)	contracts	contract_ID	6/1/2024	12/31/9999	1	NULL
32	squaremeter_size	Squaremeter size in between [0,500]	5	in_range(squaremeter, 0, 500)	housing	squaremeter	7/23/2024	9/13/2024	0	NULL

Developement of the process:
2 Validation in R

- Development of the R code in R Studio
- Collection of data and validation rules from SQL tables
- Select which data to validate by collection_IDs
- Validation
- Extraction of results into excel / SQL table

[validation].[errors]

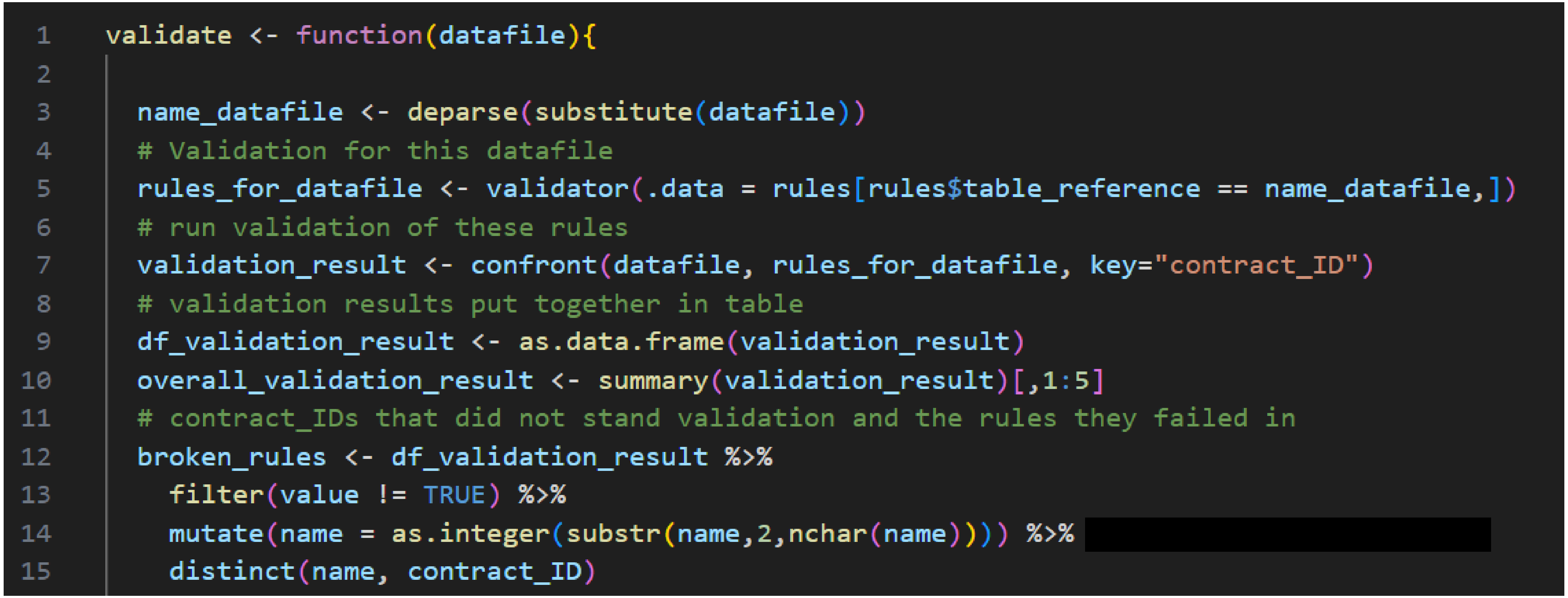


error_ID	contract_ID	rule_ID	collection_date	collection_ID
22599	614	21	26:16.0	37
22648	23463	31	26:16.0	37

Statistics Iceland

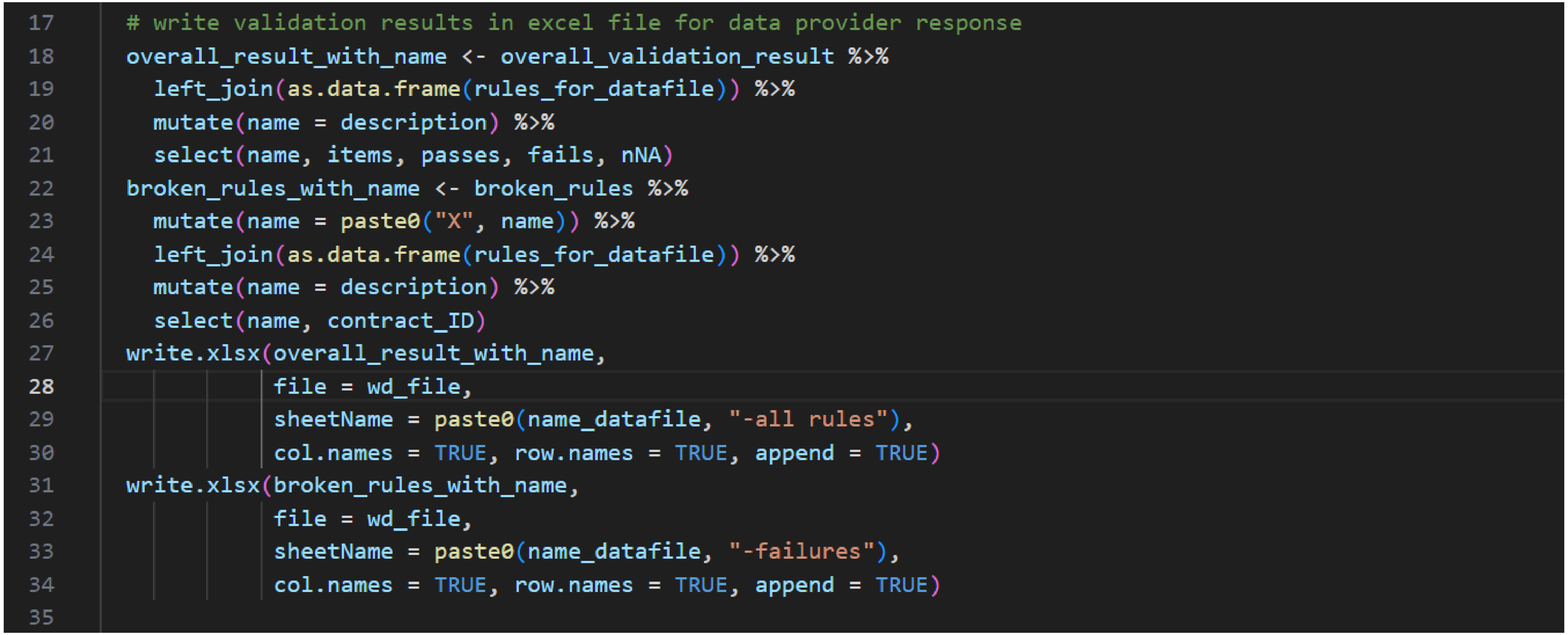
Informed society

Developement of the process:
2 Validation in R



```r
validate <- function(datafile){

  name_datafile <- deparse(substitute(datafile))
  # Validation for this datafile
  rules_for_datafile <- validator(.data = rules[rules$table_reference == name_datafile,])
  # run validation of these rules
  validation_result <- confront(datafile, rules_for_datafile, key="contract_ID")
  # validation results put together in table
  df_validation_result <- as.data.frame(validation_result)
  overall_validation_result <- summary(validation_result)[,1:5]
  # contract_IDs that did not stand validation and the rules they failed in
  broken_rules <- df_validation_result %>%
    filter(value != TRUE) %>%
    mutate(name = as.integer(substr(name,2,nchar(name)))) %>% [redacted]
    distinct(name, contract_ID)
```

Developement of the process:
3 Writing validation results into excel (Dataprovider)

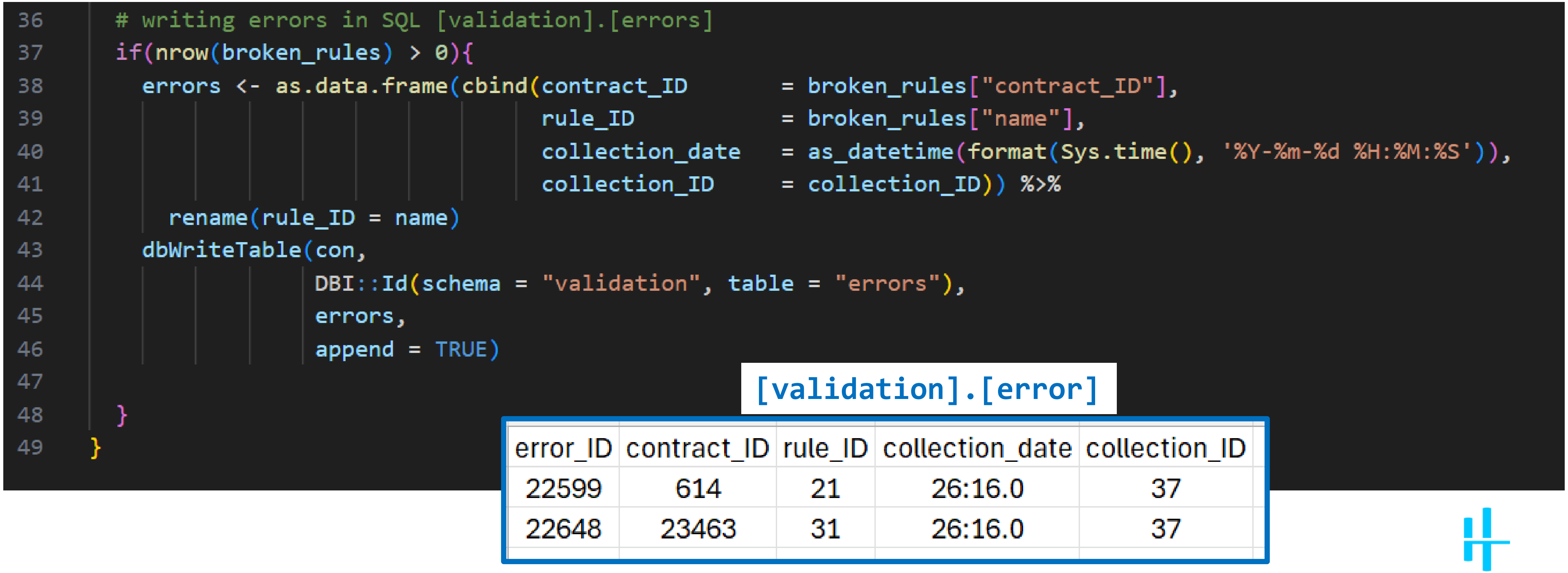


```r
# write validation results in excel file for data provider response
overall_result_with_name <- overall_validation_result %>%
  left_join(as.data.frame(rules_for_datafile)) %>%
  mutate(name = description) %>%
  select(name, items, passes, fails, nNA)
broken_rules_with_name <- broken_rules %>%
  mutate(name = paste0("X", name)) %>%
  left_join(as.data.frame(rules_for_datafile)) %>%
  mutate(name = description) %>%
  select(name, contract_ID)
write.xlsx(overall_result_with_name,
           file = wd_file,
           sheetName = paste0(name_datafile, "-all rules"),
           col.names = TRUE, row.names = TRUE, append = TRUE)
write.xlsx(broken_rules_with_name,
           file = wd_file,
           sheetName = paste0(name_datafile, "-failures"),
           col.names = TRUE, row.names = TRUE, append = TRUE)

```

Statistics Iceland

Informed society

Developement of the process:
4 Writing validation results into SQL tables



```
  # writing errors in SQL [validation].[errors]
  if(nrow(broken_rules) > 0){
    errors <- as.data.frame(cbind(contract_ID       = broken_rules["contract_ID"],
                                  rule_ID           = broken_rules["name"],
                                  collection_date   = as_datetime(format(Sys.time(), '%Y-%m-%d %H:%M:%S')),
                                  collection_ID     = collection_ID)) %>%
      rename(rule_ID = name)
    dbWriteTable(con,
                 DBI::Id(schema = "validation", table = "errors"),
                 errors,
                 append = TRUE)

  }
}
```

[validation].[error]

error_ID	contract_ID	rule_ID	collection_date	collection_ID
22599	614	21	26:16.0	37
22648	23463	31	26:16.0	37

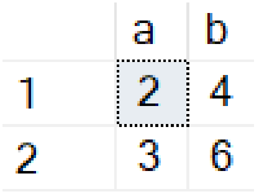
Statistics Iceland

Informed society

Transfer to SQL Stored Procedure, basic example

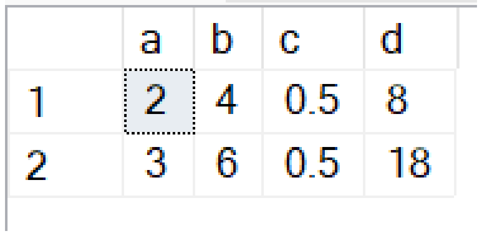
```
CREATE PROCEDURE [dbo].[CalculateValues]
AS
BEGIN
    EXEC sp_execute_external_script
         @language = N'R‘,
         @script = N‘ #library()
            a <- InputDataSet$a
            b <- InputDataSet$b
            c <- a / b
            d <- a * b
            OutputDataSet <- data.frame(a, b, c, d)',
        @input_data_1 = N'SELECT * FROM [dbo].[example];',
        @output_data_1_name = N'OutputDataSet'
    WITH RESULT SETS ((a FLOAT, b FLOAT, c FLOAT, d FLOAT));
END;

EXEC [dbo].[CalculateValues]
```

[dbo].[example]



	a	b
1	2	4
2	3	6

OutputDataSet



	a	b	c	d
1	2	4	0.5	8
2	3	6	0.5	18

Statistics Iceland

Informed society

Obstacles

- R version control

  (SQL Server 2022 Machine Learning package : R 4.2.0)

- R package management

- Language issues (Icelandic letters in the data, utf-8) on SQL Server 2019 (R 3.5.2)

- Loading in more than one SQL table per Stored Procedure – in progress

- Resource management

Pros/Cons of automation directly on SQL Server using R code within ML package

Pro

- Resssource **efficient**: Processing ressources and time
- Data is processed where it is stored
- User/Machine independent automation
- Perfect addition to fully automized data flow process
- Standardized way of data validation (reusable)
- Comparability between process qualities
- Continous data quality insurance
- Implementing of additional validation checks
- Improved accuracy in detecting data anomalies and inconsistencies
- Successful **scheduling** of validation tasks
- Potential for **Machine Learning / AI** implementation

Con

- Need for Machine Learning package on SQL Server
- Recommended from SQL Microsoft Server 2022 onwards (otherwise language issues)
- Learning threshold to implement R code into Stored Procedures
- Only one SQL table can be read in in per Stored Procedure at a time
- Installation of R packages (IT support) within ML package

THANK YOU

Do you have any questions?

Statistics Iceland

Informed society

Used R packages

- Validate[1]: for validation cran.r-project.org/web/packages/validate/
  - declare rules
  - apply them on dataset

- xlsx[2]: production of validation files to return to data providers, only if applicable cran.r-project.org/web/packages/xlsx/

- DBI [3] / odbc[4] for read in of SQL tables / writing results into SQL tables , only if not run on SQL Server cran.r-project.org/web/packages/DBI/ / cran.r-project.org/web/packages/odbc/