RAPID Methodology

Version 1.0

Created by	Reviewer	Approval
Igor Peres	Esteban Garcia	XXX
Tom Edinburg	Igor Peres	

Last revision date	Version
02 November 2025	1.0

Overview

This document outlines the analytical recommendations and structured methodologies for all ISARIC data analysis projects. Its core objective is to establish Reusable Analytical Pipelines (RAPs), which formalize and standardize the entire analytical workflow. By clearly defining and explaining concepts from Data Cleaning and Modelling through to Validation and Visualization, these RAPs enable data analysts to consistently and efficiently generate high-quality, reproducible analyses, thereby accelerating discovery in infectious disease research.

Introduction

Reusable Analytical Pipelines (RAPs) are automated statistical and analytical processes that adopt best practices from software engineering and analytical sciences. The goal of RAPs is to ensure that analyses are: Automated, Reproducible, Transparent, Maintainable, Efficient, and Robust.

The main purpose of RAPs is to improve the quality, reliability, and transparency of analysis. Specifically, RAPs aim to reduce human error by minimizing manual steps, strengthen methodological rigor, increase reproducibility and auditability of analyses, enable efficient collaboration across teams, improve long-term maintainability of research outputs, and promote transparency, accountability, and open science.

Key elements of RAPs typically involve automation to minimize manual steps and increase efficiency, the use of open-source languages such as R and Python to improve portability and reduce license costs, peer code review to enhance quality and reliability, version control tools like Git for auditability, change tracking, and disaster recovery, external publication of code and methods to promote transparency and reuse, and thorough documentation to ensure methods and workflows are understandable and reproducible.

1. The Audience

The RAPID methodology, built on the principles of Reusable Analytical Pipelines (RAPs), is designed to support a broad spectrum of users engaged in research and data analysis. These users bring diverse expertise and needs, which shapes how they interact with the methodology and its tools. To ensure the approach remains accessible, effective, and extensible, two primary audiences are considered:

1) **End-users of the RAPID package**: This group includes researchers, analysts, and domain experts who apply RAPID tools to conduct automated, reproducible, and transparent analyses. They are primarily focused on generating insights and evidence using established workflows, without necessarily engaging in software development. These users benefit from clear documentation, intuitive interfaces, and robust analytical outputs that align with best practices in scientific research.
2) **Contributors and developers**: This audience comprises data scientists, software engineers, and technical researchers who seek to extend RAPID's capabilities. Their contributions may include developing new functionalities, refining existing components, or integrating RAPID with other platforms and methodologies. These users rely on access to source code, version control systems, and collaborative environments such as GitHub to support innovation and continuous improvement.

It is also recognized that many users may discover RAPID through varied channels—such as academic events, training sessions, peer recommendations, or online searches—and may not fit neatly into predefined categories. The methodology is therefore designed to be flexible and inclusive, enabling both adoption and contribution regardless of entry point or technical background.

2. ISARIC Approach

The **International Severe Acute Respiratory and Emerging Infection Consortium (ISARIC)** is a global federation of investigator-led clinical research networks, with more than 70 members worldwide. ISARIC conducts clinical research to improve patient care and to facilitate a globally coordinated and agile research response to infectious disease threats. ISARIC is committed to generating and disseminating timely and relevant clinical research evidence wherever outbreaks occur.

2.1. RAPID: Reusable Analytical Pipelines for Infectious Diseases

Building on the RAP concept, ISARIC has adapted the idea into **RAPID** (*Reusable Analytical Pipelines for Infectious Diseases*), tailored to the specific needs of clinical research on emerging infectious diseases. Every **RAPID** is designed to solve research questions about the **clinical characterisation of infectious diseases**. They primarily use data collected through ISARIC ARC-type CRFs[1], though their design accommodates all ARC-formatted data[2].

RAPID pipelines are intended to support analyses that:

- Characterise the clinical presentation of cases, including natural history, progression, and variability of symptoms
- Enable comparative clinical epidemiology across diseases, populations, and geographies
- Identify risk factors and outcomes associated with disease
- Describe patient management practices and heterogeneity
- Summarise diagnostic approaches and results
- Inform case management strategies, support clinical trial design, and guide public health interventions

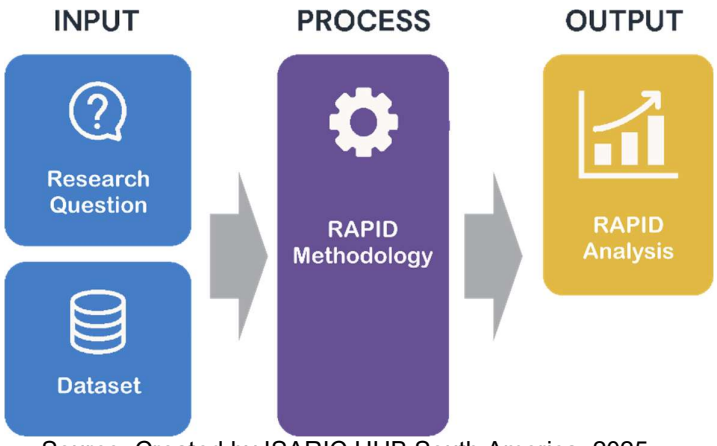
In this context, Reusable Analytical Pipelines for Infectious Diseases (RAPID) represent a structured methodology that integrates data science techniques to address clinical research questions with transparency, consistency, and reproducibility. Built upon the expertise of the ISARIC team, the RAPID framework offers a step-by-step guide to support researchers in answering specific clinical questions using optimal predictive models.

The RAPID methodology follows a structured analytical flow: it begins with a clinical research question and a curated dataset as inputs. These are processed through a reproducible pipeline of

[1] CCP research question - https://isaricresearch.github.io/CCP/CCP_toolkit
[2] CCP research question - https://isaricresearch.github.io/CCP/CCP_toolkit

data science techniques. The output is a transparent and validated analysis that answers the research question with clinical relevance, as shown in Figure 1.

Figure 1 – RAPID methodology flow.



Source: Created by ISARIC HUB South America, 2025.

This structured approach not only enhances the reproducibility and transparency of clinical analyses but also streamlines decision-making by guiding researchers through a consistent and validated process. By aligning data inputs with analytical rigor, the RAPID methodology ensures that clinical questions are addressed with precision, enabling scalable insights across infectious disease research

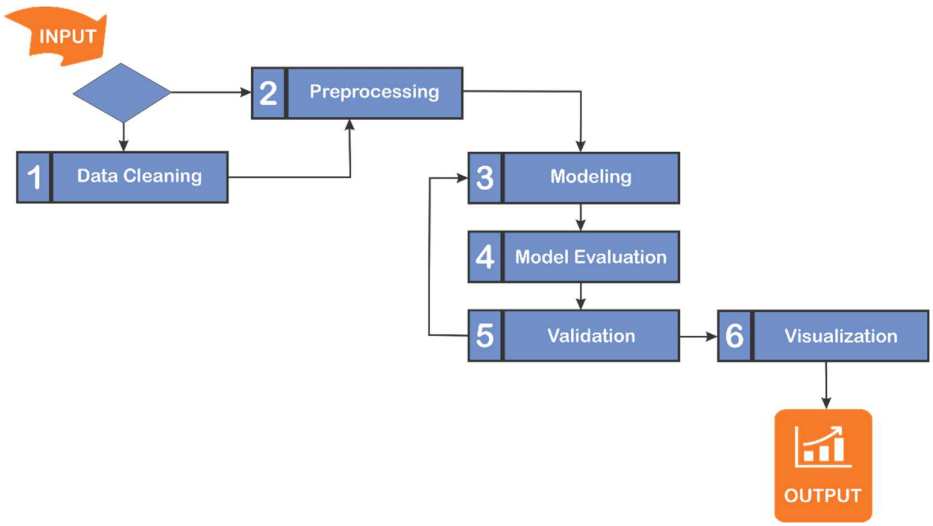
2.2. Analytical Process in RAPID

A **RAPID** is defined as a **pathway across selected methods within the six analytical steps**, designed to answer a specific clinical research question. Importantly, both data and associated metadata should pass through a RAPID. The starting datesets are patient-level data, in a flexible data schema linked to ARC that includes both wide-format and long-format records. Intermediate processed data frames should be accompanied by additions or updates to the metadata. The output of a RAPID must be aggregated data frames that do not contain patient-level information. Visualization is performed only on these aggregated outputs, ensuring patient data protection.

By this definition, **multiple RAPID pipelines can exist for the same research question** (for example, risk factor analysis conducted via logistic regression versus random forest). Each RAPID pipeline is **reusable, adaptable, and transparent**, enabling consistent replication across different diseases, populations, and outbreak contexts.

All analytical procedures within RAPID pipelines follow **six core steps,** as shown in the proposed methodology (Figure 2). Each step has a selection of methods. The user may choose which method to apply, based on the research question.

Figure 2 - Methodology of Reusable Analytical Pipelines for Infectious Diseases



Source: Created by ISARIC HUB South America, 2025.

In the following a description of each of existing phases of the methodology are explained, considering what is the phase? Why is the phase important? And what is the output of the end of the phase?

2.2.1. Data Cleaning

Data cleaning is the initial, critical phase to remove errors and inconsistencies in raw data before performing modelling or analysis. The presence of errors or inconsistencies can lead to biased or incorrect conclusions, as many statistical and machine learning methods are sensitive to the quality of the input data.

It involves identifying and correcting issues such as duplicate records, incompatible formats, entry errors, or missing values. This step is essential because it ensures the reliability of the dataset. Without it, subsequent analyses may be affected by distortions, noise, or misleading interpretations.

By the end of the cleaning process, the data are ready to be transformed and analysed with greater precision, allowing the following stages - such as preprocessing and modelling - to be carried out more effectively and securely.

Although data cleaning is the first phase of the RAPID methodology, it is not mandatory if the dataset has already undergone a rigorous cleaning process. In such cases, researchers may proceed directly to preprocessing and modeling steps. However, it is recommended to verify the integrity and consistency of the data before skipping this phase, especially when working with external or previously processed datasets.

2.2.2. Data Preprocessing

The stage following data cleaning, involving a series of transformations to prepare the features for modelling and analysis. This step ensures that the data meets the assumptions of the chosen algorithm and often significantly improves model performance.

This step enhances the interpretability and reliability of results by addressing potential inconsistencies and adapting the data structure as needed.

Depending on the context, adjustments may include scaling, transform variables, or select relevant information. By the end of this stage, the dataset is structured to support robust and meaningful statistical or computational modelling.

2.2.3. Modeling

Applying statistical or machine learning methods to the pre-processed data to answer research questions, make predictions, or discover patterns.

It is crucial because the choice of model and its configuration directly impact the accuracy and generalizability of the results. Different algorithms (e.g., regression, decision trees, neural networks) may be tested depending on the nature of the data and the research objective.

By the end of this phase, one or more candidate models are generated, each capable of producing predictions based on input data, ready for evaluation and refinement.

2.2.4. Model Evaluation

The process of assessing the performance and robustness of the developed modelling approach. This involves calculating various metrics on independent data to quantify the model's predictive ability.

This phase is essential to determine how well the model generalizes to unseen data and whether it meets the clinical relevance criteria. Poorly evaluated models may lead to misleading conclusions or unsafe recommendations.

The output of this phase is a set of performance indicators that guide the selection of the most suitable model for the research question, balancing predictive power and interpretability.

2.2.5. Validation

The process of confirming the findings and assessing the generalisability of the model beyond the immediate development data.

This step is critical to ensure that the model is not overfitted to the training data and can be trusted in broader clinical contexts. It strengthens the credibility and reproducibility of the findings.

The result of the validation phase is a robust, generalizable model with documented performance across diverse scenarios, ready to support clinical decision-making or further research.

Durante the modeling, evaluation, and validation phases, iterative refinement is not only possible but encouraged. As models are trained and assessed, researchers may identify limitations, biases, or performance gaps that require adjustments in feature selection, algorithm choice, or parameter tuning. The RAPID framework supports this feedback loop, allowing analysts to return to previous steps, improve model robustness, and ensure that the final output is both accurate and clinically meaningful. This iterative flexibility is essential for producing reliable insights in complex clinical datasets.

2.2.6. Visualization

The process of presenting results clearly and effectively for interpretation, communication, and clinical decision-making.

It is important because visual outputs enhance understanding among clinicians, researchers, and stakeholders, bridging the gap between data science and clinical practice.

At the end of this phase, the analysis is presented in a clear and accessible format, supporting evidence-based conclusions and enabling informed decisions.

2.3. RAPID Design Decisions

To implement **RAPID** in a way that is both practical and sustainable for ISARIC's global clinical research network, we have made the following design choices:

- **Automation:** RAPID pipelines will prioritise automation wherever feasible to minimise manual steps and improve efficiency. However, **analytical decisions**, executed in the form

of choices of parameters for specific methods, must be designed according to the context in which the pipeline is applied. RAPID pipelines are built on the **ISARIC data schema**:

- **ARC-formatted data:** For datasets captured using **ARC CRF question structure** and the **Clinical Characterisation REDCap XML**, which provide standardised data capture structures, fully automated transformation methods will be available.
- **Non-ARC structured data:** For datasets that deviate from the ARC structure, whether in CRF question format or data capture structure, a **semi-automatic auto-mapper process** will be applied to align data with the ISARIC schema.

- **Open-source languages:** All pipelines are implemented in **Python**, chosen for its widespread adoption, extensive analytical libraries, and integration with clinical research workflows, al also how it links with other tools that novel machine learning tools, and visualization and web development tools
- **Code review:** Code is reviewed collaboratively through **GitHub** and within the **ISARIC analytical community**, ensuring quality, reliability, and peer accountability.
- **Version control:** All code is tracked and managed using **GitHub version control**, enabling full auditability, traceability of changes, and disaster recovery.
- **External publication:** RAPID code and workflows will be shared through open **GitHub** repositories. Outputs and methods may also be disseminated through **preprints** or **white papers**.
- **Documentation:** Documentation is a core requirement. Each RAPID pipeline will include:
  - **Code-level documentation** (docstrings, comments, structure)
  - **Tutorials and walkthroughs** hosted on GitHub
  - **User guides** explaining the assumptions, steps, and outputs of each pipeline
- **Coding standards/checklist for contributions (tbc)**
  - DRY principle (i.e. don't repeat yourself)
  - Conforms to consistent style (where possible, follow existing RAPIDs), including PEP 8 and PEP 257 style guides
  - This includes consistency of inputs/outputs so that methods can be easily exchanged when appropriate (e.g. logistic regression to random forest)
  - Each pipeline must have unit and integration testing, and end-to-end tests involving openly-available (synthetic?) example datasets
  - Package dependencies should be limited as much as possible to established and well-maintained packages, and unused dependencies should be removed
  - The GitHub repository should have standard structure, and should contain README.md, details about installation, tutorials, CONTRIBUTING.md,
  - E.g. (but for discussion):
    - https://gist.github.com/tomschr/c23585f6bcbeb6374f183deb83352f36
    - https://peps.python.org/pep-0008

- https://google.github.io/styleguide/pyguide.html

3. RAPIDs (Reusable Analytical Pipelines for Infectious Diseases)

This chapter covers a recommended analytical path within the RAPID methodology, guiding researchers from the formulation of a clinical research question to the generation of validated and interpretable results. The chapter also highlights iterative refinement opportunities and suggests software packages methods to support reproducibility and modular implementation. By following this structured approach, researchers can ensure methodological consistency and analytical transparency in infectious disease studies.

3.1. Risk Factor Analysis

A risk factor is a characteristic of an individual (e.g., demographics, a biomarker, a previous health condition) that is associated - causally or not - to a certain outcome. The risk factor can be interpreted as a predictor of a certain outcome or has an etiologic relationship with it. For instance, in certain health conditions, if an individual has hypertension then the probability of he or she is having heart disease is high.

Identifying a risk factor can be done with different statistical modeling approaches. For example, one can estimate the statistical correlation between two variables - the condition and the outcome. In most cases, a risk factor is interpreted as a predictor variable of a certain outcome (response variable). Hence, regression modeling is widely used since it provides an easy and straightforward way to model relationships among a response variable and one or more predictor variables, which also may present some interrelationships that impact the estimation of their individual effect (or association).

3.1.1. Methodology

To conduct a risk factor modeling or analysis, one can use the following steps:

1. State the hypothesis of the analysis: the first step is to set the outcome of interest and the risk factor candidates. To set the hypothesis, previous knowledge can be used to list the risk factors already known or discussed in literature or clinical practice. For instance, according to previous studies, the presence of high fever, including thrombocytopenia or bleeding are risk factors for severe Dengue, hence, new studies should account for these variables and estimate their association with the outcome in different settings.

2. Data collection: Once a hypothesis is stated and potential risk factors candidates are listed, they should be measured. Different data collection methods could be used either if the collection is retrospective or prospective.

3. Data preparation: After a dataset is obtained, it should be prepared before modeling, especially with respect to regression analysis. A few points are worth noticing when preparing data and are described below (For more information, see the Data preparation guide)

    a. Inconsistency of values: Values may be inconsistent due to errors in the data collection process (e.g., input errors) or values that are out of the common range of a variable's distribution. In these cases, correction of values should be preferred, however, if that is not an option, values can be set to missing and treated with other methods.

    b. Correlated variables: Characteristics that are statistically correlated represent similar information for modeling and may represent unnecessary increased variance, thus impacting the inference of a risk factor's effect. Hence, variables should be modeled to reduce or remove correlation: a single variable can be selected or variables can be combined with the correlated variables using simple mathematical modeling or sophisticated techniques such as clustering or Principal Component Analysis (PCA). Correlation can be assessed using the Pearson's (linear) or Spearman's (nonparametric) correlation coefficient for two numerical variables; the Chi-square test and Cramer's V correlation for two categorical variables, and t-test (parametric) or Mann-Whitney U test (nonparametric) for one categorical and one numerical variable.

    c. Normalization: For some types of models, it is necessary to normalize the variables in the same scale. Min-max normalization will be used when this preparation is necessary.

    d. Missing values: Correspond to data that was not collected; hence, the value is absent from the dataset. Missing value patterns may differ depending on the root cause of its missingness, hence, the pattern of missing values should be evaluated and identified (e.g., missing completely at random [MCAR], missing at random [MAR], missing not at random [MNAR]). For each pattern, different methods for imputation can be used. In some cases, observations with missing values can be removed from the dataset ("complete cases" approach). Tests available for MCAR are Little test (parametric) and Jamshidian and Jalal's test (non-parametric). Assessing the influence of missing data through descriptive analysis and visualizations is also informative. A recommended strategy for MCAR is to use MICE (Multiple Imputation by Chained Equations), a statistical method for handling missing data by iteratively estimating missing values, considering relationships between variables, and generating multiple imputed datasets to account for uncertainty. For further information, please check the Missing data guide.

e. Feature selection: To create parsimonious models, it is recommended to select the most important features to include in the model. Lasso and Recursive Feature Elimination are two approaches used to evaluate the best set of features to be selected. For further information, please check the Variable selection guide.

4. Modeling: Risk factors modeling consist in representing response variables (e.g. the outcome) as a mathematical function of a set of predictor variables (e.g., characteristics, health conditions, etc.). The common approach is estimating a linear function using regression analysis, however, different functions can be hypothesized and estimated. A “risk factor” is identified as a variable which increases the risk of an outcome. Linear regression models assume that the outcome is a numerical variable defined in the interval [-inf, +inf] of a normal distribution and is a function of the linear combination among predictors. However, the outcomes can be different in terms of nature (binary, categorical or numerical) and, therefore, the modelling may vary according to the statistical behavior.

   a. Binary outcomes: If the response variable is a binary outcome (e.g., 1 - death and 0 - alive), one of the common models used to analyze such data is the logistic regression model. This model estimates the probability of obtaining one of the two possible outcome values (e.g., 1) as a function of a set of predictor variables through a logit transformation. Specifically, the probability of the event occurring is modeled as the logistic (sigmoid) function applied to a linear combination of the predictors, ensuring that the predicted values always fall within the (0,1) range. In this framework, the estimated coefficients of the predictors represent changes in the log-odds of the outcome per unit increase in the predictor variable. Since the log-odds scale can be difficult to interpret directly, researchers often exponentiate these coefficients, obtaining the odds ratio (OR). The odds ratio quantifies the multiplicative change in the odds of the outcome occurring for each unit increase in the predictor. An OR greater than 1 indicates that the predictor is associated with higher odds of the event occurring, while an OR less than 1 suggests a protective effect or decreased odds. Logistic regression also allows for adjustments for confounders and interactions among predictors, making it a powerful tool for analyzing binary outcomes in various fields, including epidemiology, social sciences, and machine learning.

   b. Numerical outcomes: If the response variable is a numerical countable (discrete) variable, it is often assumed to follow a distribution from the exponential family, such as the Poisson or Negative Binomial distribution. These models are commonly used to analyze count data, where the outcome represents the number of occurrences of an event within a fixed unit of observation (e.g., hospital visits per year, number of infections per patient). In these cases, the expected count rate is modeled as a logarithmic function of a linear combination of predictor variables, ensuring that the predicted values remain non-negative. The estimated coefficients in these models represent changes in the

logarithm of the expected count rate for each unit increase in the corresponding predictor. To facilitate interpretation, researchers often exponentiate these coefficients, yielding the rate ratio (RR), which quantifies the multiplicative change in the expected count per unit increase in the predictor. A rate ratio greater than 1 indicates an increased event occurrence, while a rate ratio below 1 suggests a decrease in the event frequency. Additionally, if the outcome is naturally a rate rather than a simple count (e.g., number of infections per 1,000 patient-days), the model can include an offset variable, which represents the exposure time or population at risk. The offset is incorporated with a fixed coefficient of 1, effectively adjusting the model to account for different observation periods or population sizes. This approach ensures that the estimated effects correspond to differences in event rates rather than raw counts, making the model more applicable to real-world scenarios in epidemiology, public health, and actuarial science.

c. Recently, supervised machine learning models for classification or regression, such as Random Forest, Support Vector Machines (SVMs), and Neural Networks, have been increasingly used as alternatives to traditional statistical models. These models offer high predictive accuracy and flexibility, as they can capture complex, nonlinear relationships between predictors and outcomes without requiring strong parametric assumptions. However, a key challenge with these methods is the interpretability of the results, which is key in risk factors analysis. Unlike traditional models like logistic or linear regression, where coefficients have clear and direct meanings, machine learning models are often considered "black boxes," making it difficult to assess the importance or direction of the association of individual variables with the outcome. To address this issue, methods from Explainable Artificial Intelligence (XAI) have been developed to provide insights into how these models make predictions. One of the most widely used techniques is the SHapley Additive exPlanations (SHAP) values, which are based on cooperative game theory. SHAP values quantify the contribution of each predictor to a given prediction by distributing the difference between the actual prediction and a baseline value among the input features. This allows researchers to interpret the relative importance of variables, detect nonlinear effects, and even uncover potential biases in the model. Other popular explainability techniques include Partial Dependence Plots (PDPs), Local Interpretable Model-Agnostic Explanations (LIME), and Accumulated Local Effects (ALE), each offering different perspectives on feature importance and model behavior. Further information on Machine Learning methods can be found in the ML and Predictive Modelling guides

5. Model diagnostics: In risk factor analysis, validating models is a crucial step to ensure their reliability and robustness. One of the primary methods for model validation is residual evaluation, which involves assessing the characteristics of the residuals (the difference between observed and predicted values). Residual analysis helps detect potential model

misspecifications, such as non-linearity, heteroscedasticity (non-constant variance), and outliers, which can compromise the model's predictive performance and interpretability. In addition to residual diagnostics, performing a functional analysis of each numerical predictor is considered a best practice. This step is essential because many statistical models, such as linear and logistic regression, assume a linear relationship between predictors and the outcome. If this assumption is violated, transformation techniques (e.g., polynomial terms, splines, or logarithmic transformations) can be applied to better capture complex associations. Furthermore, information criteria metrics, such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), are widely used to evaluate and compare model fitness. These metrics balance model goodness-of-fit with model complexity, penalizing the inclusion of excessive parameters to prevent overfitting. Lower AIC or BIC values indicate a better trade-off between model performance and parsimony. Beyond these methods, other techniques such as cross-validation, likelihood-ratio tests, and predictive accuracy metrics (e.g., area under the ROC curve for classification or R^2 for regression) can also be used to further validate model robustness. By integrating these diagnostics, researchers can ensure that their models are not only statistically sound but also generalizable and interpretable in real-world applications.

6. Risk factor identification: In statistical modeling, each predictor variable may have an association with the response variable, representing its estimated effect on increasing or decreasing the probability or magnitude of the outcome. Understanding these associations is fundamental in epidemiology, clinical research, and risk assessment studies. In regression models, the effects of predictors are quantified by their coefficients in the linear combination of variables. These coefficients indicate the expected change in the outcome per unit increase of the predictor, assuming all other variables remain constant. To assess the uncertainty and reliability of these estimates, confidence intervals (CIs) are commonly reported. A narrow CI suggests a more precise estimate, while a wide CI indicates greater variability and uncertainty. When the CI does not include the null value (e.g., 0 in linear regression or log-odds regression, 1 in odds ratios or rate ratios), the association is considered statistically significant. A risk factor is characterized by a positive effect on the outcome, meaning it increases the probability or level of the response variable. Conversely, protective factors have a negative effect, decreasing the likelihood or magnitude of the outcome. These effects are critical in identifying modifiable risk factors in public health interventions and clinical decision-making. To facilitate the interpretation of results, the estimated effects can be presented in tables or graphical visualizations such as forest plots. Forest plots are particularly useful in meta-analyses and multivariable regression models, as they provide a clear summary of effect sizes and confidence intervals across different predictors. Other visualization techniques, such as partial dependence plots (PDPs) and SHAP summary plots, can further enhance the understanding of complex relationships in machine learning-based models.

3.1.2. RAPID Code implementation

Step 1 – Data Cleaning

In this initial step, we apply the remove_outliers method to eliminate extreme values that may distort the analysis. Outliers can significantly affect survival models, especially when estimating hazard ratios or survival curves. By removing them, we improve the reliability and stability of the modeling process.

Function call:

- isa.datacleaning.remove_outliers(data, method="iqr")

Step 2 – Preprocessing

Preprocessing prepares the dataset for modeling by handling missing values, encoding categorical variables, and scaling continuous features. We use the MICE technique for imputation, which is well-suited for clinical datasets with multivariate missingness. This step ensures that the input data is complete and standardized.

Function calls:

- isa.preprocessing.mice(data_cleaned, strategy="pmm", max_iter=5)
- isa.preprocessing.encode_categoricals(data_imputed)
- isa.preprocessing.normalize(data_encoded)

Step 3 – Modeling

For survival analysis, we use the Cox Proportional Hazards model, which estimates the effect of covariates on the time to event. This model is widely used in clinical research due to its interpretability and ability to handle censored data.

Function call:

- isa.modeling.coxph(data_scaled, time_column="time_to_event", event_column="event")

Step 4 – Model Evaluation

To evaluate the model's performance, we calculate the Concordance Index and Brier Score. These metrics assess how well the model predicts survival times and the accuracy of probabilistic predictions.

Function calls:

- isa.evaluation.concordance_index(model)
- isa.evaluation.brier_score(model)

Step 5 – Validation

Validation is performed using bootstrapping, which helps assess the model's stability and generalizability. This technique resamples the data multiple times to estimate the variability of the model's performance.

Function call:

- isa.validation.bootstrap(model, n_iterations=1000)

Step 6 – Visualization

To communicate results effectively, we generate survival curves and hazard ratio plots. These visualizations help interpret the model's predictions and the influence of covariates on survival outcomes.

Function calls:

- isa.visualization.survival_curves(model)
- isa.visualization.hazard_ratios(model)

Python Code:

```
import isaric as isa

if __name__ == "__main__":
    # Step 1: Data Cleaning (optional)
    data_cleaned = isa.datacleaning.remove_outliers(data, method="iqr")
    data_cleaned = isa.datacleaning.handle_duplicates(data_cleaned)
```

```
    data_cleaned = isa.datacleaning.standardize_formats(data_cleaned)

    # Step 2: Preprocessing
    data_imputed = isa.preprocessing.mice(data_cleaned, strategy="pmm", max_iter=5)
    data_encoded = isa.preprocessing.encode_categoricals(data_imputed)
    data_scaled = isa.preprocessing.normalize(data_encoded)

    # Step 3: Modeling (Survival Analysis)
    model = isa.modeling.coxph(data_scaled, time_column="time_to_event",
event_column="event")

    # Step 4: Model Evaluation
    concordance_index = isa.evaluation.concordance_index(model)
    brier_score = isa.evaluation.brier_score(model)

    # Step 5: Validation
    validation_results = isa.validation.bootstrap(model, n_iterations=1000)

    # Step 6: Visualization
    isa.visualization.survival_curves(model)
    isa.visualization.hazard_ratios(model)
```

3.1.3. Tutorials and Examples

A set of tutorials and examples of risk factor-related analysis the ISARIC Clinical and Epidemiology platform's tools and functions are available. Implementations are currently in Python and uses the ISARIC Analytics and ISARIC Draw packages (<>)

a. Linear regression;

b. Logistic regression;

c. Survival Analysis using Cox Regression;

Any issues and bugs, please report to data@isaric.org or the Github (https://github.com/ISARICResearch)