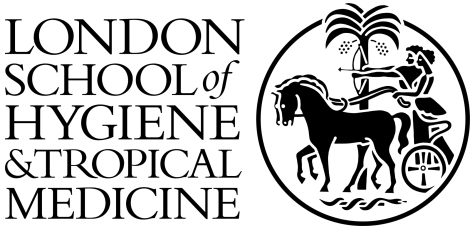


Epiverse interoperability: data standards and structures

Joshua W. Lambert

Epiverse-Harmonize Workshop

July 2023

Why use data standards?

Routine analyses can utilise consistent and robust data objects to enable interoperability

Bespoke data structures can be optimised to tasks at hand (e.g., ease of use, memory management, performant data manipulation)

Well-defined data structures can be validated to check for structural, numerical, statistical correctness

Object-oriented programming in R

Functional

S3

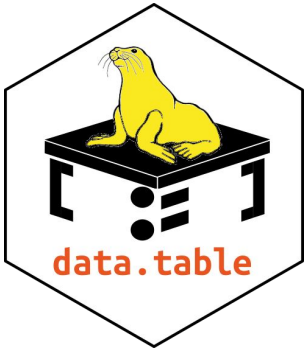
S4

Encapsulated

Data Wrangling in R

Packages for data cleaning/processing

General packages



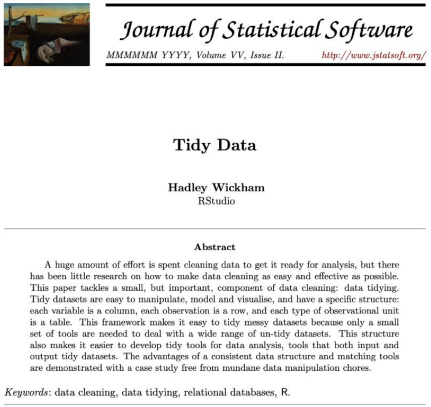
Domain-specific packages:

Epi: {incidence2}

Climate: {ClimActor}

Bioinformatics: {POMA}

Principles for data processing



Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. http://www.jstatsoft.org/

Tidy Data

Hadley Wickham
RStudio

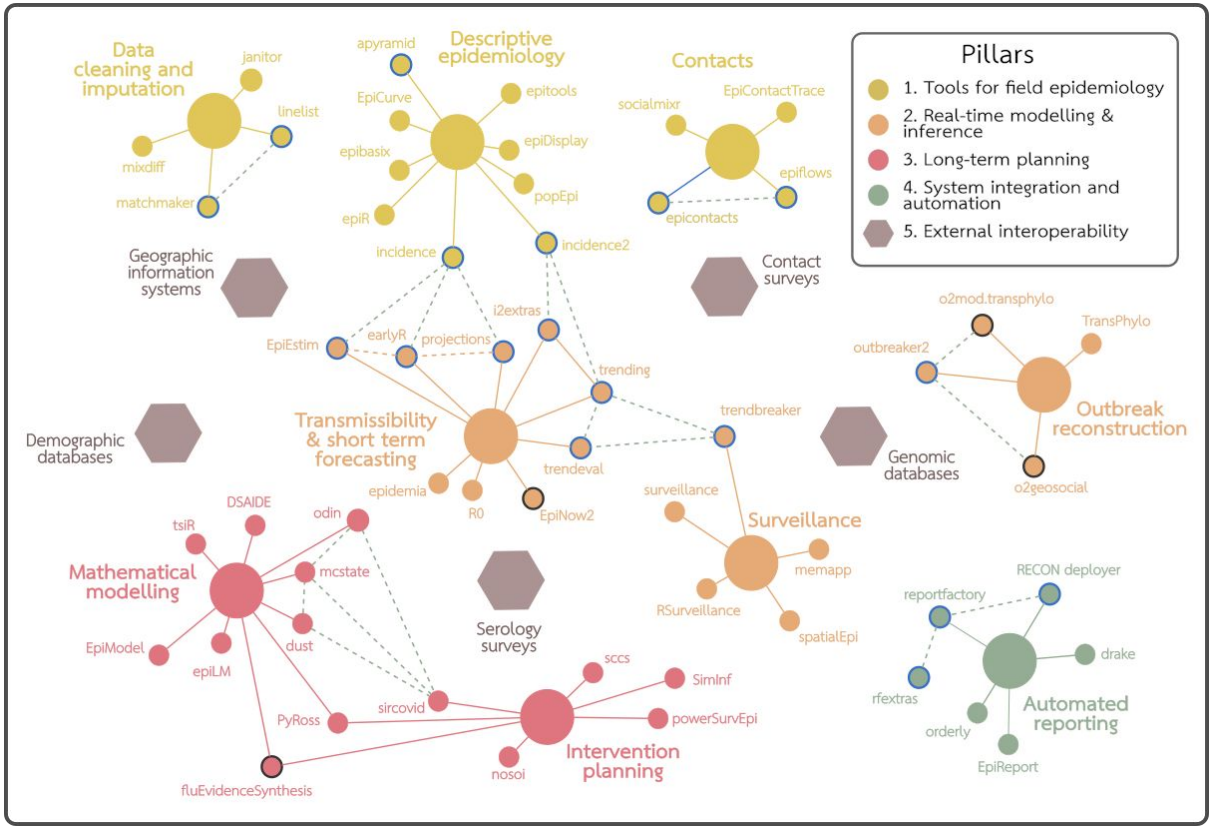
Abstract

A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible. This paper tackles a small, but important, component of data cleaning: data tidying. Tidy datasets are easy to manipulate, model and visualise, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table. This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.

Keywords: data cleaning, data tidying, relational databases, R.

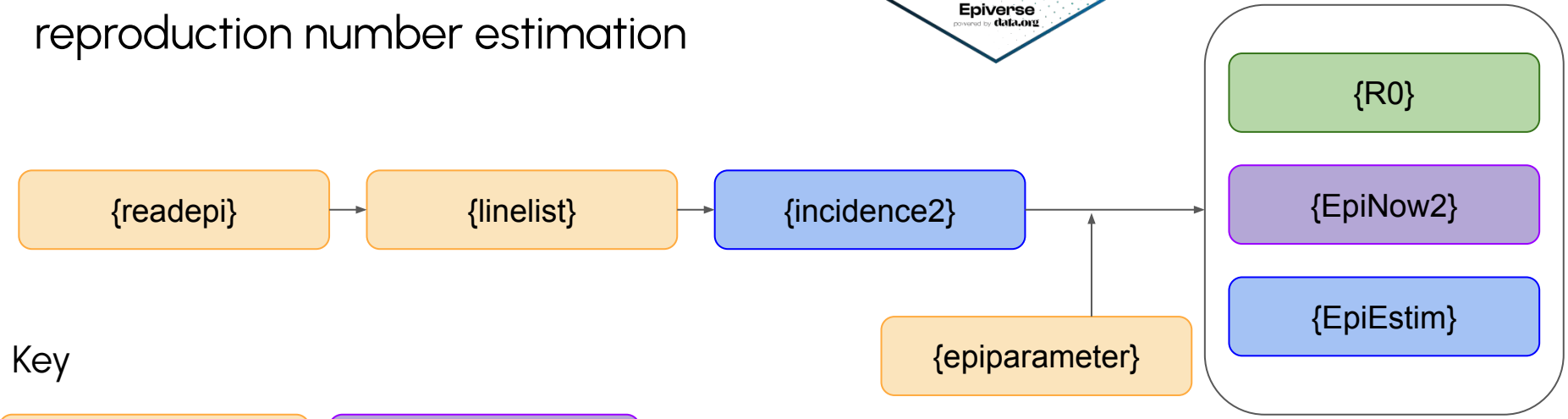
Wickham (2014)

Existing tooling



Epiverse analytics pipeline

Transmissibility data pipeline for reproduction number estimation



Key

Epiverse-TRACE R package

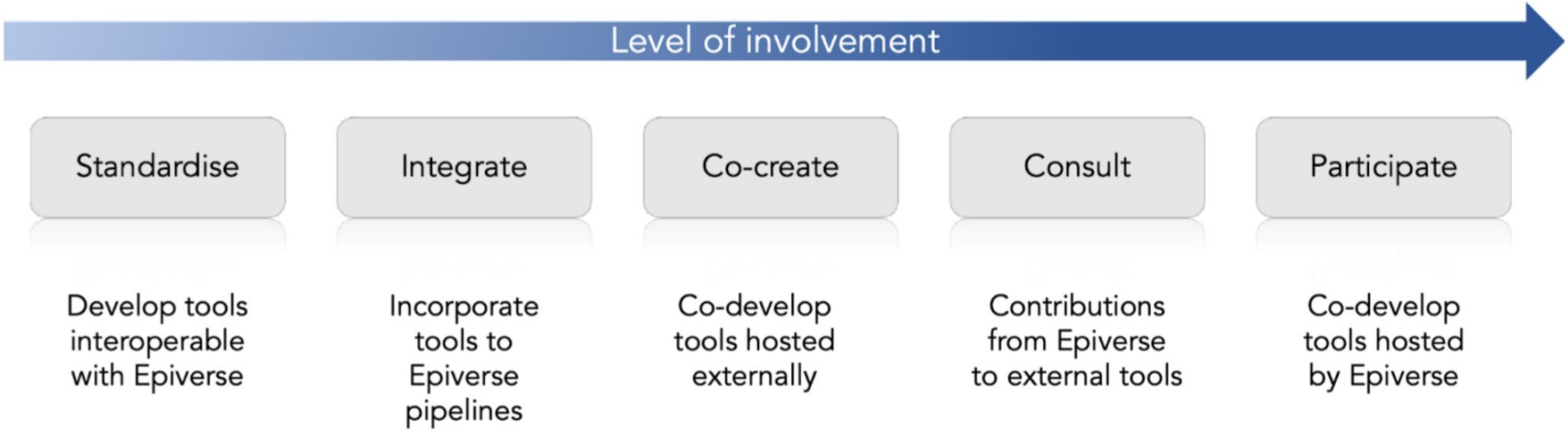
RECON R package

External (close collaboration)

External

The Epiverse community

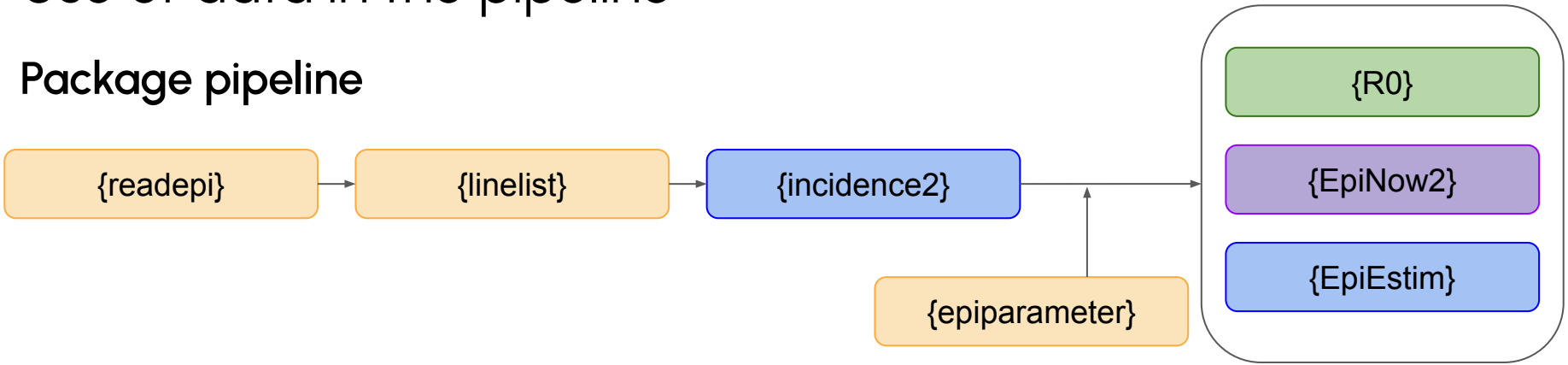
"Epiverse-TRACE aims to support the development of integrated, generalisable and scalable community-driven software for epidemic analytics, and contribute to a sustainable ecosystem of existing and new tools."



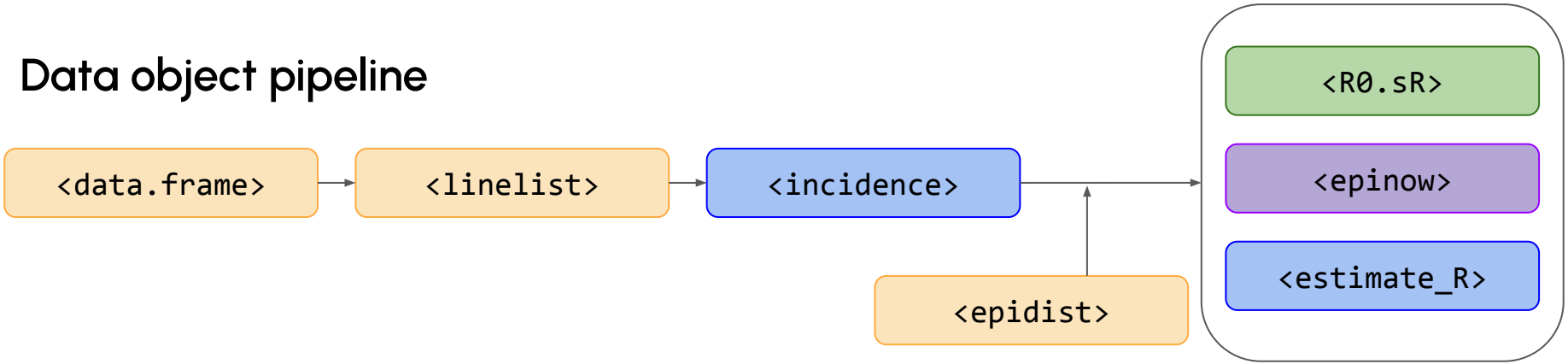
https://epiverse-trace.github.io/

Use of data in the pipeline

Package pipeline



Data object pipeline



Epiverse choices

S3 OOP used by Epiverse thus far.

Pros

- Familiar usage and syntax
- Relatively easy to write and maintain

Cons

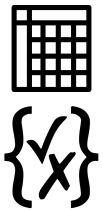
- No type checking

Caveat:

Run-time validation checking with validator functions (class invariants)

Static validation

- Data dictionary
- JSON validation



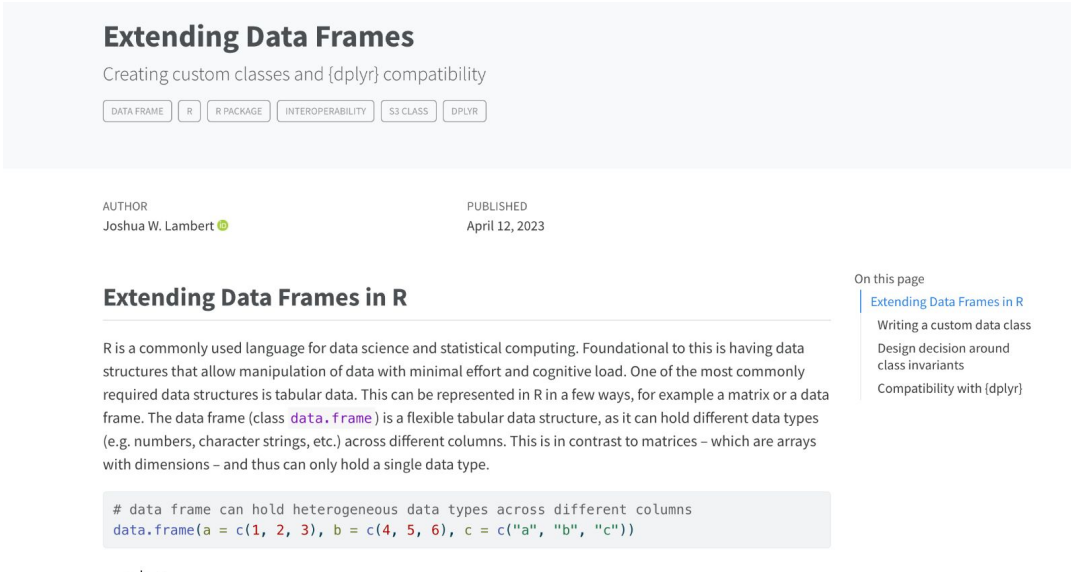
Building classes

Creating S3 class

```r
var <- list(a = 1, b = 2)
class(var) <- "new_class"
# or
var <- structure(
  list(a = 1, b = 2),
  class = "new_class"
)
```

Inheriting from existing S3 class

```r
var <- data.frame(a = 1, b = 2)
class(var) <- c("new_class", "data.frame")
```



Extending Data Frames

Creating custom classes and {dplyr} compatibility

DATA FRAME | R | R PACKAGE | INTEROPERABILITY | S3 CLASS | DPLYR

AUTHOR
Joshua W. Lambert

PUBLISHED
April 12, 2023

Extending Data Frames in R

On this page
- Extending Data Frames in R
- Writing a custom data class
- Design decision around class invariants
- Compatibility with {dplyr}

R is a commonly used language for data science and statistical computing. Foundational to this is having data structures that allow manipulation of data with minimal effort and cognitive load. One of the most commonly required data structures is tabular data. This can be represented in R in a few ways, for example a matrix or a data frame. The data frame (class `data.frame`) is a flexible tabular data structure, as it can hold different data types (e.g. numbers, character strings, etc.) across different columns. This is in contrast to matrices – which are arrays with dimensions – and thus can only hold a single data type.

```r
# data frame can hold heterogeneous data types across different columns
data.frame(a = c(1, 2, 3), b = c(4, 5, 6), c = c("a", "b", "c"))
```

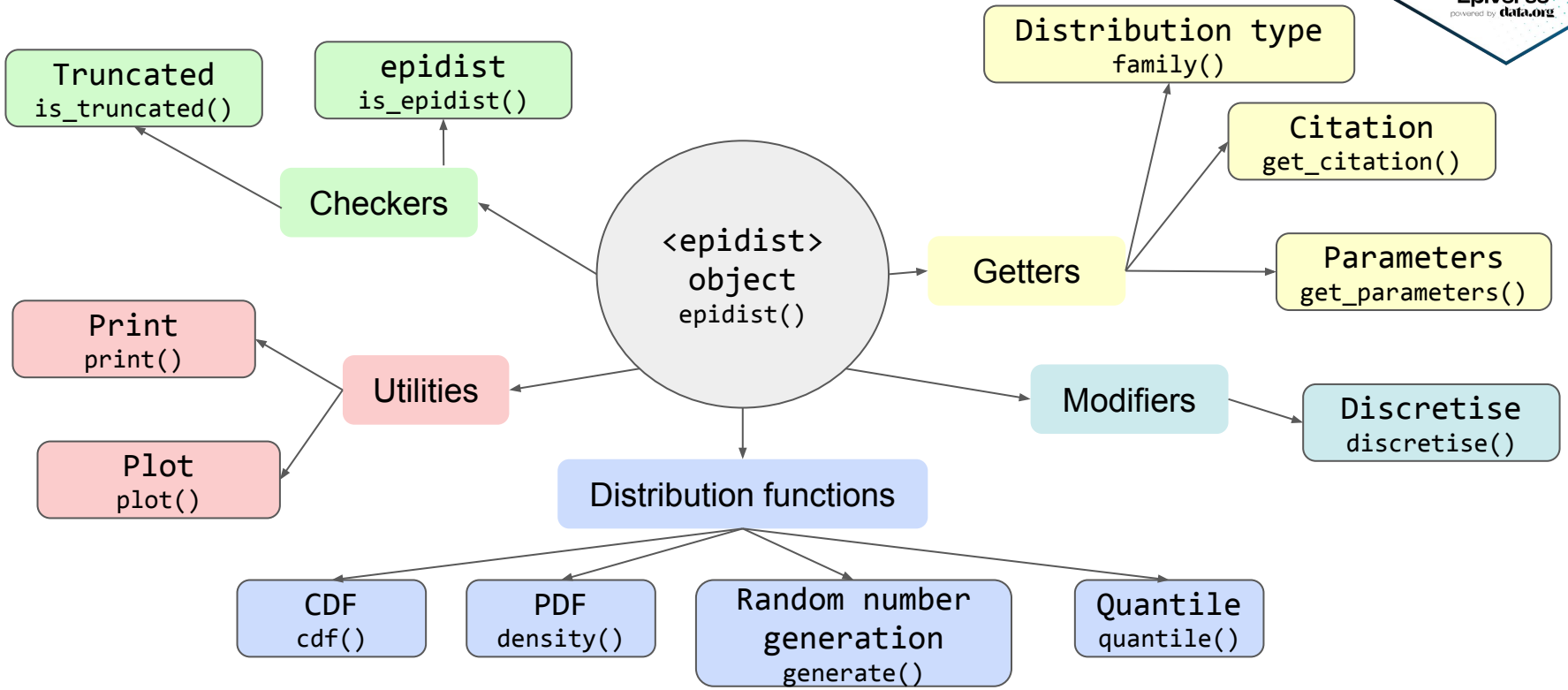
https://epiverse-trace.github.io/posts/extend-dataframes/

Challenges

New objects can seem esoteric and may put off new users

Need to make these objects intuitive to use and make the benefits of OOP clear

Object ecosystem



Other Epiverse examples

<scenarios>

Description: "The 'scenario' class is intended to store the outcomes of a number of runs of an epidemic simulation"

<infection>
<vaccination>
<population>
<intervention>

Simple APIs & "Hidden Interoperability"

Function definition

```r
proportion_transmission <- function(R,
                                    k,
                                    percent_transmission,
                                    sim = FALSE,
                                    ...,
                                    epidist) {
  # function body...
}
```

Standard use

```r
proportion_transmission(
  R = 1.5,
  k = 0.6,
  percent_transmission = 0.8
)
```

Interoperability use

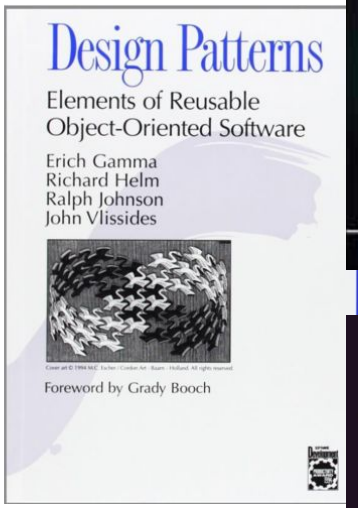
```r
# load <epidist> from {epiparameter}
proportion_transmission(
  epidist = epidist,
  percent_transmission = 0.8
)
```

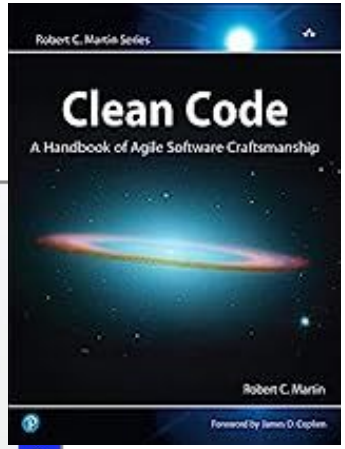
When not to write custom data objects

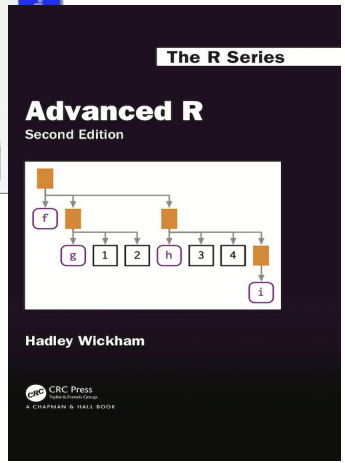
Using OOP and data standardisation is not a replacement for good design

Simple built in data types should be used when complexity is minimal

Don't reinvent the wheel; reuse established data structures







Epiverse-Harmonize opportunities

Climate-dependent epidemiological parameters

- {epiparameter} ⟷ Harmonize

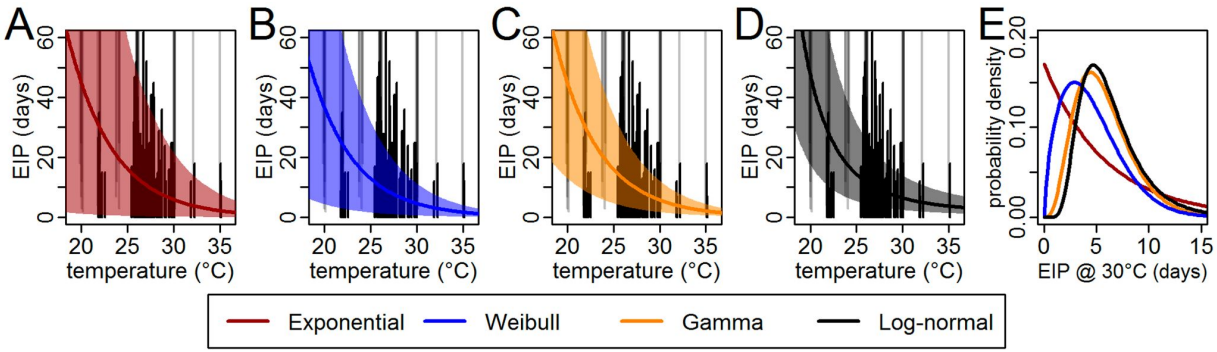
Classes for joint climate-epidemiological data

- Aggregate by time and space across data sets
- Validate data

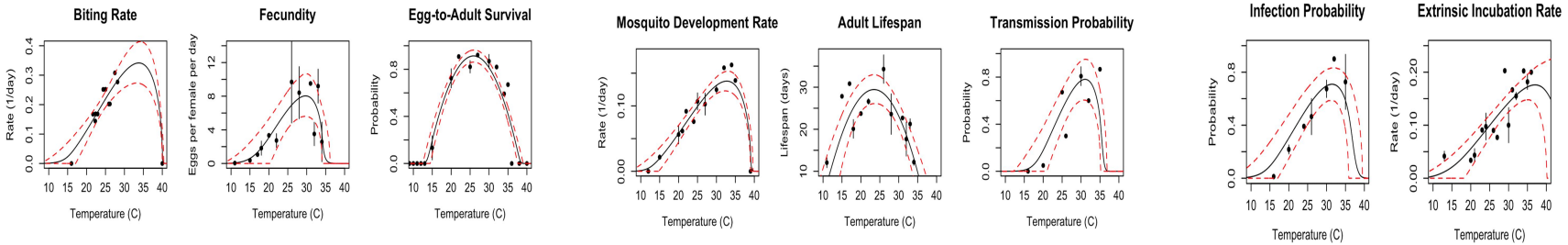
Climate-dependent processes in compartmental models

- {epidemics} ⟷ Harmonize

Climate-dependent epidemiological parameters



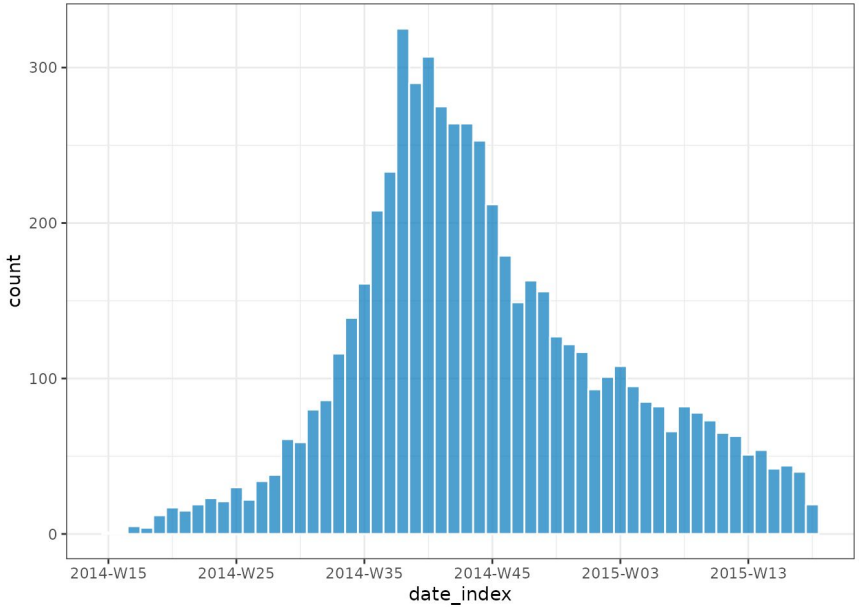
Chan and Johnson (2012)
PLOS One



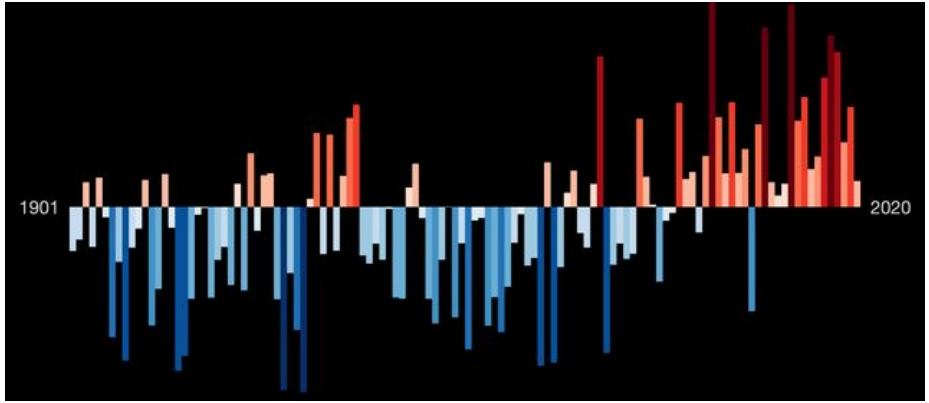
Mordecai et al. (2017) PLOS
Negl. Trop. Dis.

Spatio-temporal alignment and validation

Case data



Climate data



Thanks for listening

Any Questions?