Liesel in Statistical Research

Model Building, Simulation Studies, Technical Details

Johannes Brachem

Gianmarco Callegar

Thomas Kneib

Hannes Riebl

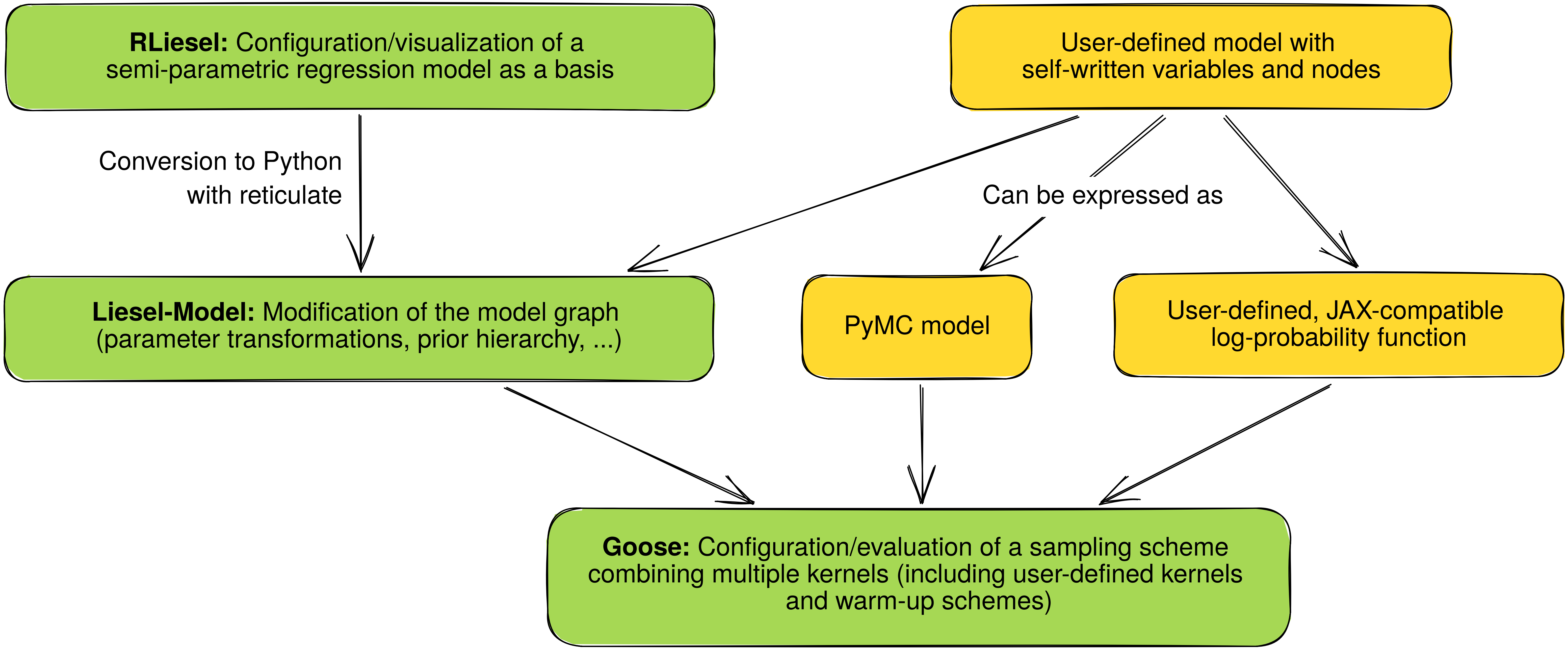
Paul F.V. Wiemann

Contents

1. Liesel in statistical research
    - Workflows
    - Case studies
2. Using Liesel…
    - … with Conda
    - … with Quarto
    - … for simulation studies
3. Exercises

Liesel in statistical research

Workflows with Liesel



RLiesel: Configuration/visualization of a semi-parametric regression model as a basis

User-defined model with self-written variables and nodes

Conversion to Python with reticulate

Can be expressed as

Liesel-Model: Modification of the model graph (parameter transformations, prior hierarchy, ...)

PyMC model

User-defined, JAX-compatible log-probability function

Goose: Configuration/evaluation of a sampling scheme combining multiple kernels (including user-defined kernels and warm-up schemes)

Case study 1: bivariate normal responses

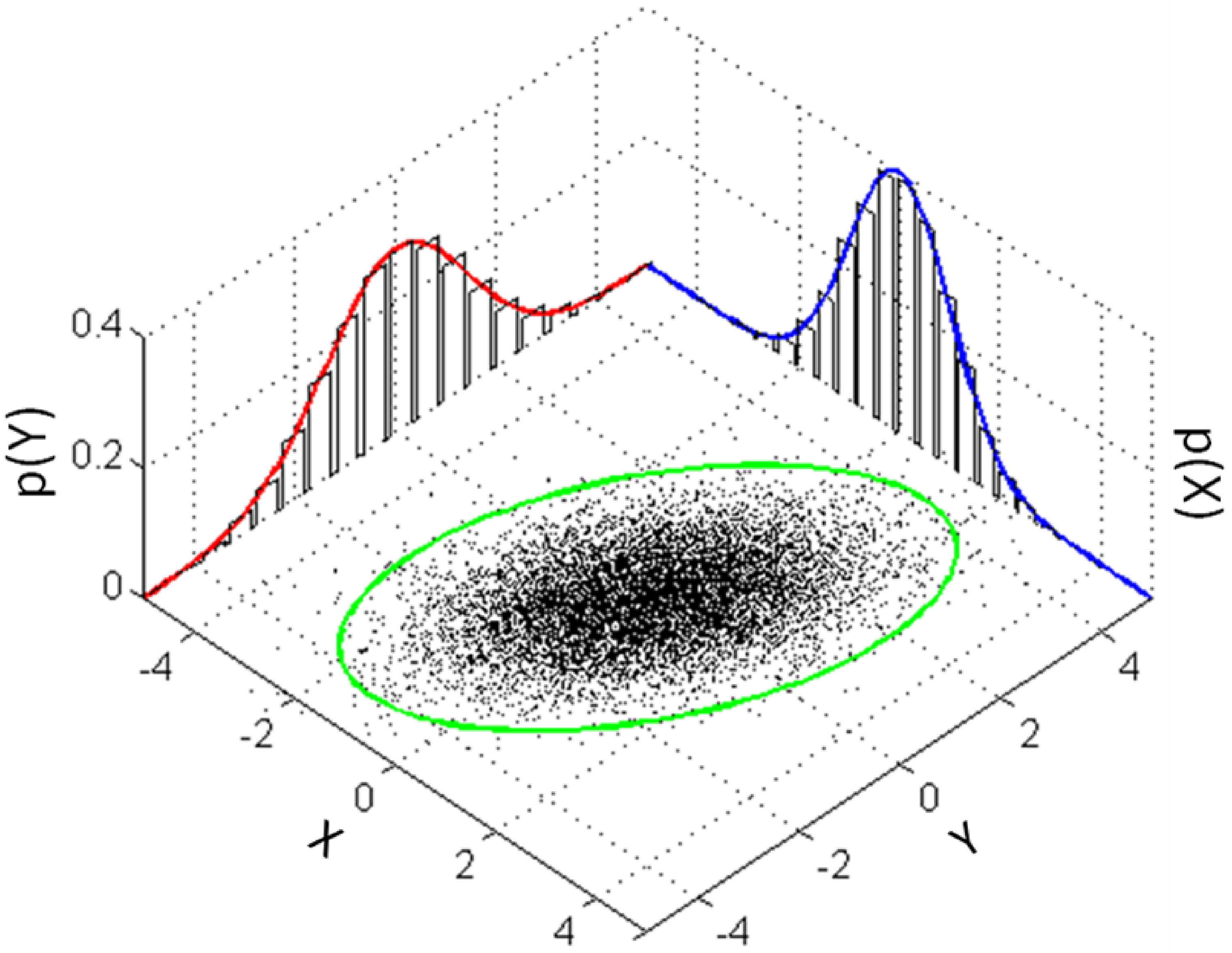


Illustration of the bivariate normal distribution (from Wikipedia)

Case study 1: bivariate normal responses (cont'd)

Parameterization of a possible regression model:

Response parameter	Predictor and response function
First marginal mean	$\mu_1 = \eta_1$
First marginal standard error	$\sigma_1 = \exp(\eta_2)$
Second marginal mean	$\mu_2 = \eta_3$
Second marginal standard error	$\sigma_2 = \exp(\eta_4)$
Correlation	$\rho = \tanh(\eta_5)$

Case study 1: bivariate normal responses (cont'd)

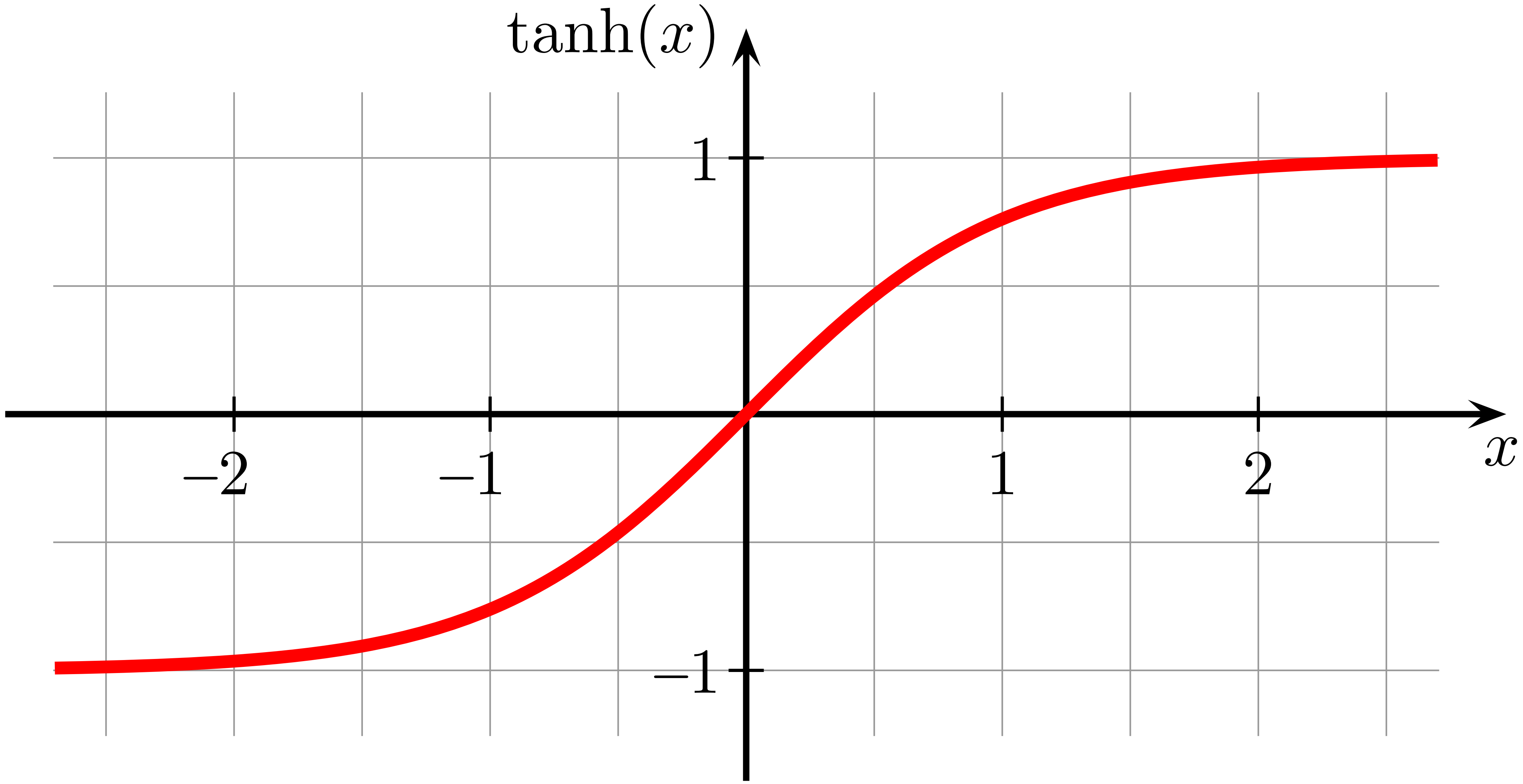


Illustration of the tanh response function (from Wikipedia)

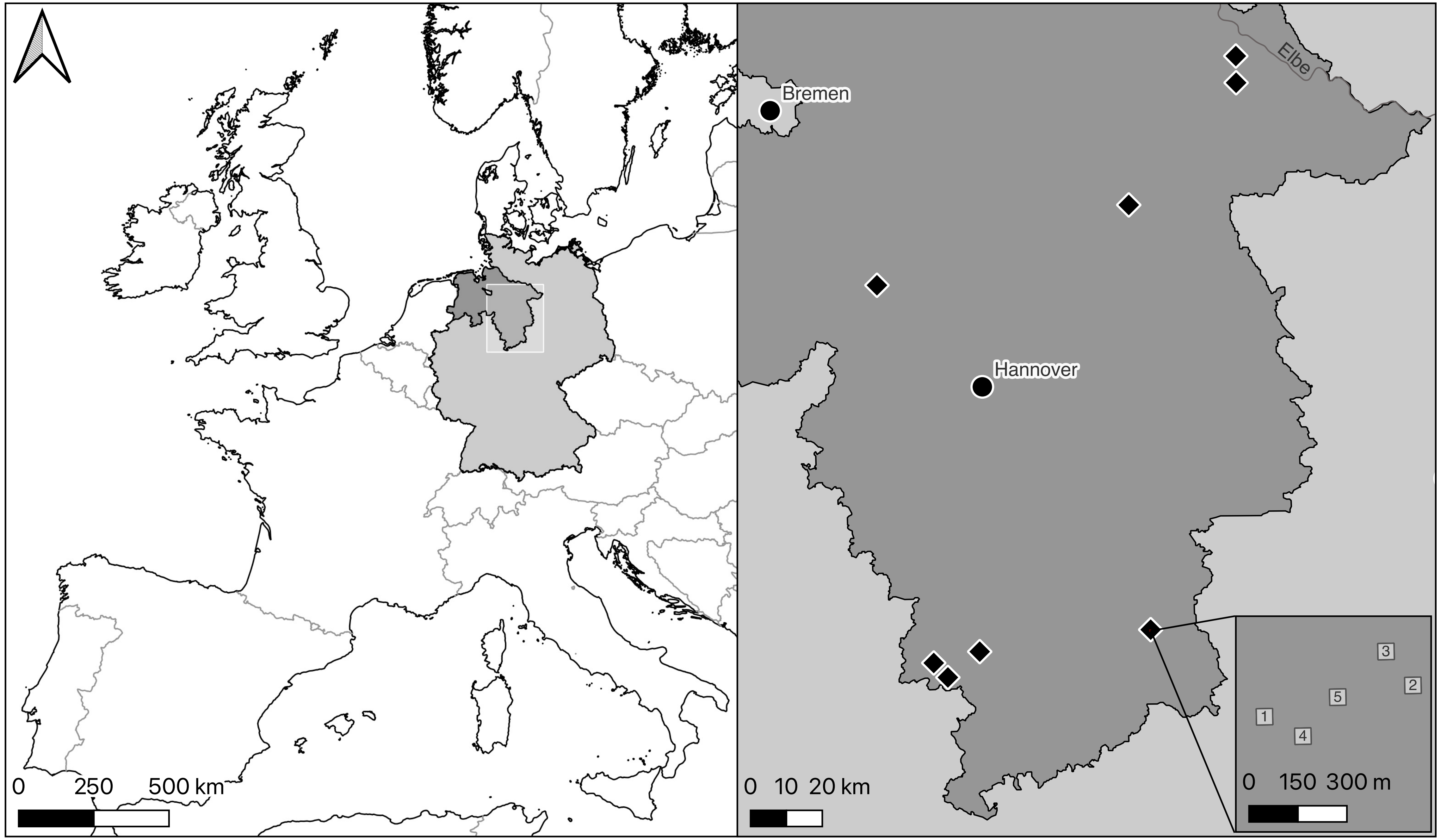
Case study 1: bivariate normal responses (cont’d)

Subclassing the TFP MVN distribution:

```
class BivariateNormal(tfd.MultivariateNormalTriL):
    def __init__(
        self, loc1=0.0, loc2=0.0, scale1=1.0, scale2=1.0, cor=0.0, ...
    ):
        ...                         # build mean vector and covariance matrix
        super().__init__(loc, scale_tril, ...)  # initialize mvn distribution
        ...

    @classmethod
    def _parameter_properties(cls, ...):
        return {
            "loc1": parameter_properties.ParameterProperties(
                default_constraining_bijector_fn=lambda: tfb.Identity(),
            ),
            ...
            "scale1": parameter_properties.ParameterProperties(
                default_constraining_bijector_fn=lambda: tfb.Exp(),
            ),
            ...
            "cor": parameter_properties.ParameterProperties(
```

(From our template repository (more later))

Case study 2: multi-species count model



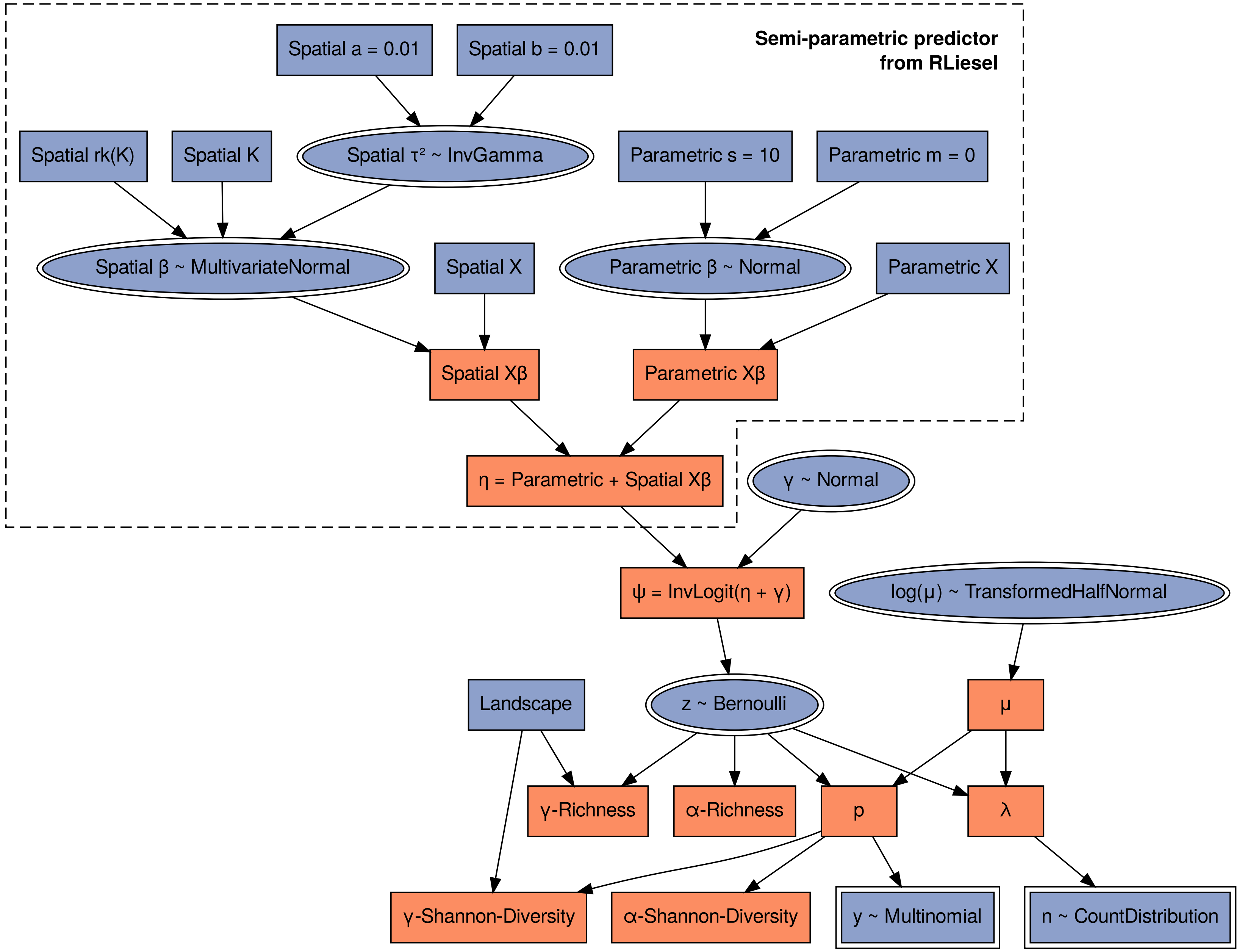
Experimental design of the Research Training Group 2300

Case study 2: multi-species count model (cont’d)

Example data for the multi-species count model:

```
> y[1:10, 1:4]
     ceratophysella_denticulata folsomia_manolachi paratullbergia_callipygos parisotoma_notabilis
1.1                           2                 31                         2                    1 ...
1.2                           3                 10                        11                    1 ...
1.3                           2                  7                         1                   21 ...
1.4                           0                 12                         1                    3 ...
1.5                           1                 79                         3                    9 ...
2.1                           6                  0                         0                   13 ...
2.2                           0                  0                         0                   12 ...
2.3                           2                  0                         0                    3 ...
2.4                           0                  0                         0                    5 ...
2.5                           0                  0                         4                    0 ...
                            ...                ...                       ...                  ...
```

Case study 2: multi-species count model (cont’d)



Graph of the model variables

Case study 2: multi-species count model (cont'd)

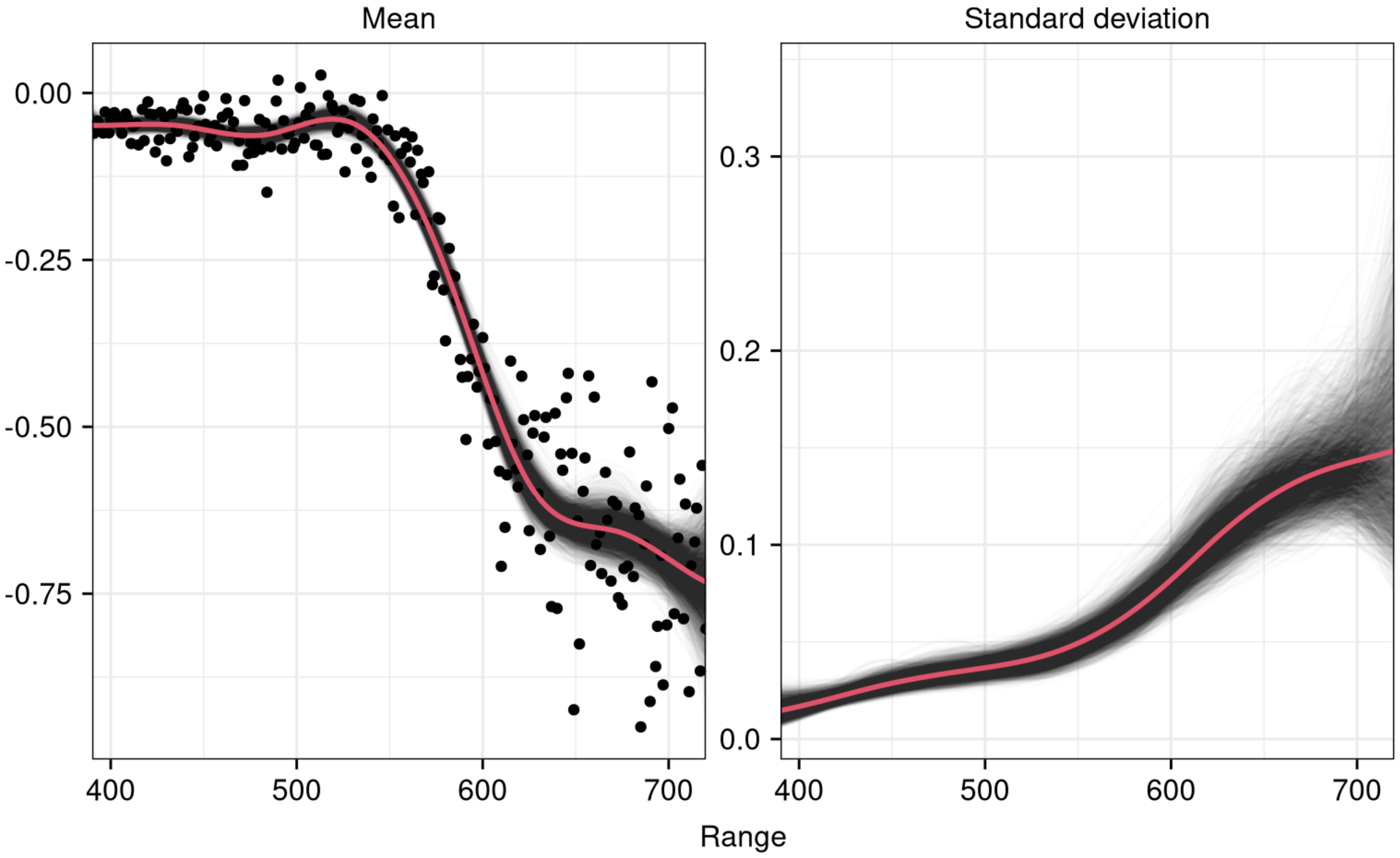
Configuring the **regression predictor** in R:

```r
gb <- liesel(..., builder = TRUE)                                      # graph builder with the regression predictor
```

Combining it with the **response structure** in Python:

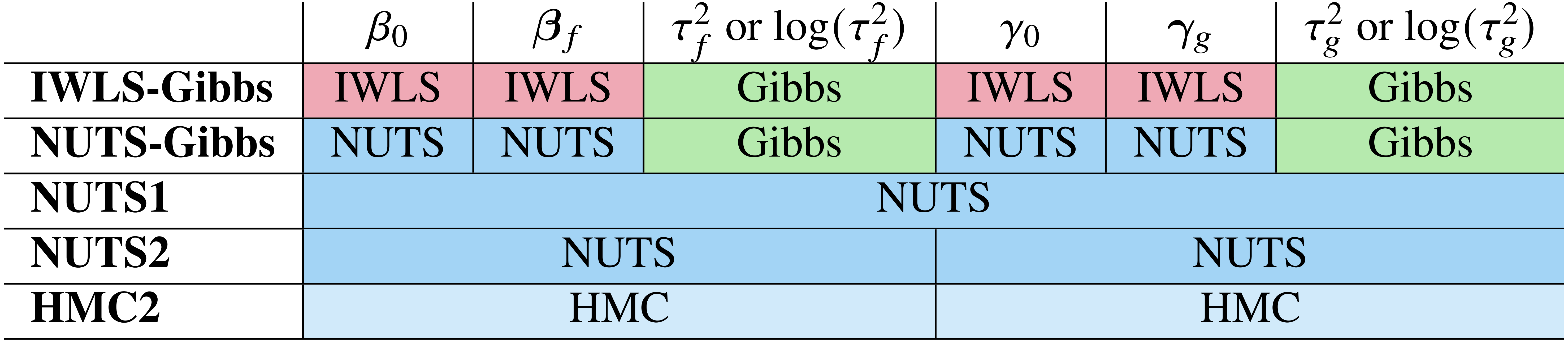
```python
pdt_vars = {var.name: var for var in r.gb.vars}                                        # extract predictor variables

model = mscm.make_model(...)                                                # model with the mscm response structure
_, mscm_vars = model.pop_nodes_and_vars()                                               # extract response variables

gb = lsl.GraphBuilder().add(                                               # add root variables to new graph builder
    mscm_vars["n"], mscm_vars["y"], mscm_vars["richness"], mscm_vars["shannon"],
    mscm_vars["richness_landscape"], mscm_vars["shannon_landscape"]
)

gb.transform(mscm_vars["mu"], tfb.Exp)                                                          # transform parameter
gb.replace_var(mscm_vars["eta"], pdt_vars["eta_pdt"])                                       # replace dummy predictor
model = gb.build_model()                                                                     # build combined model
```

Case study 3: comparing samplers



Gaussian location-scale regression model

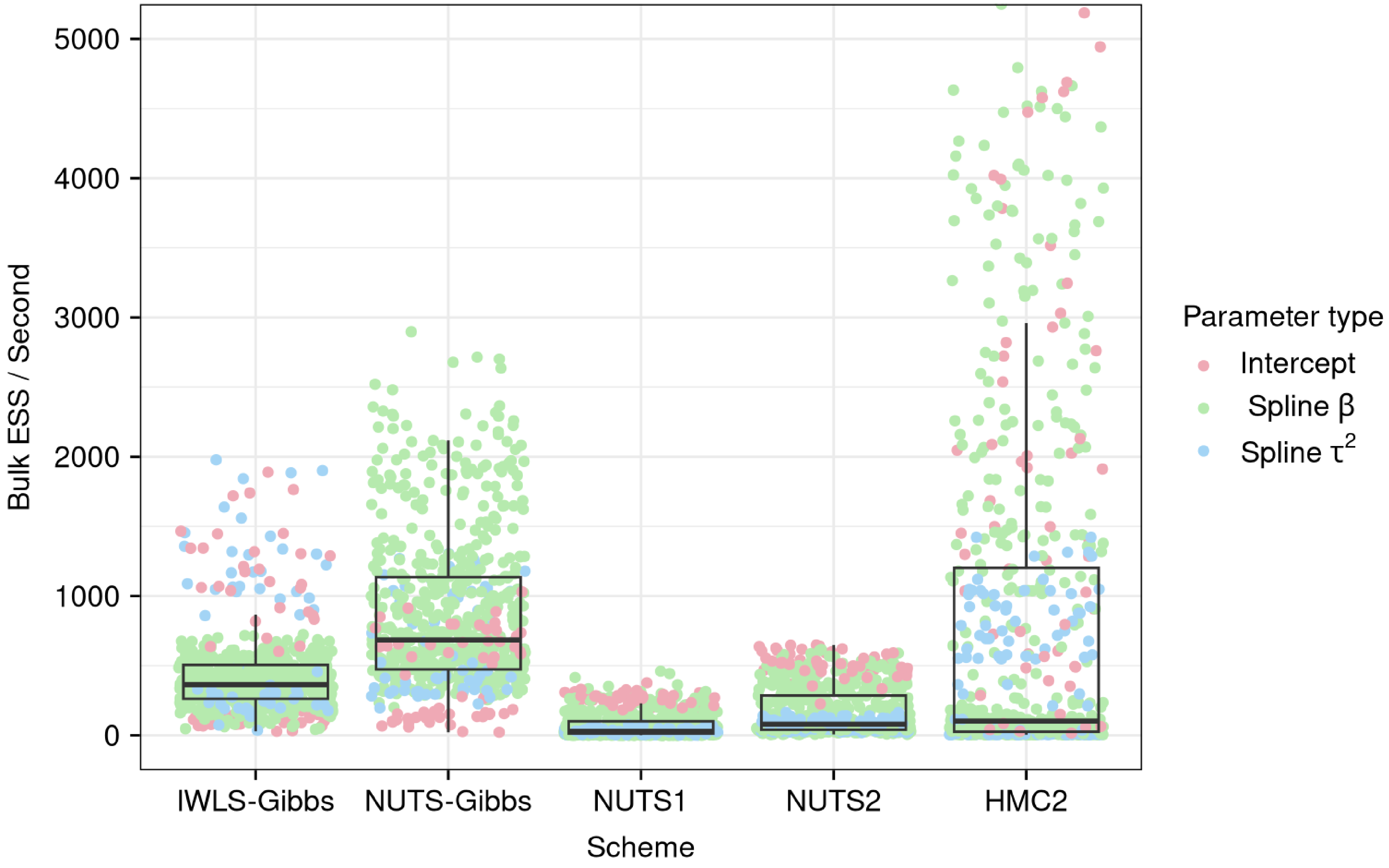
Case study 3: comparing samplers (cont'd)



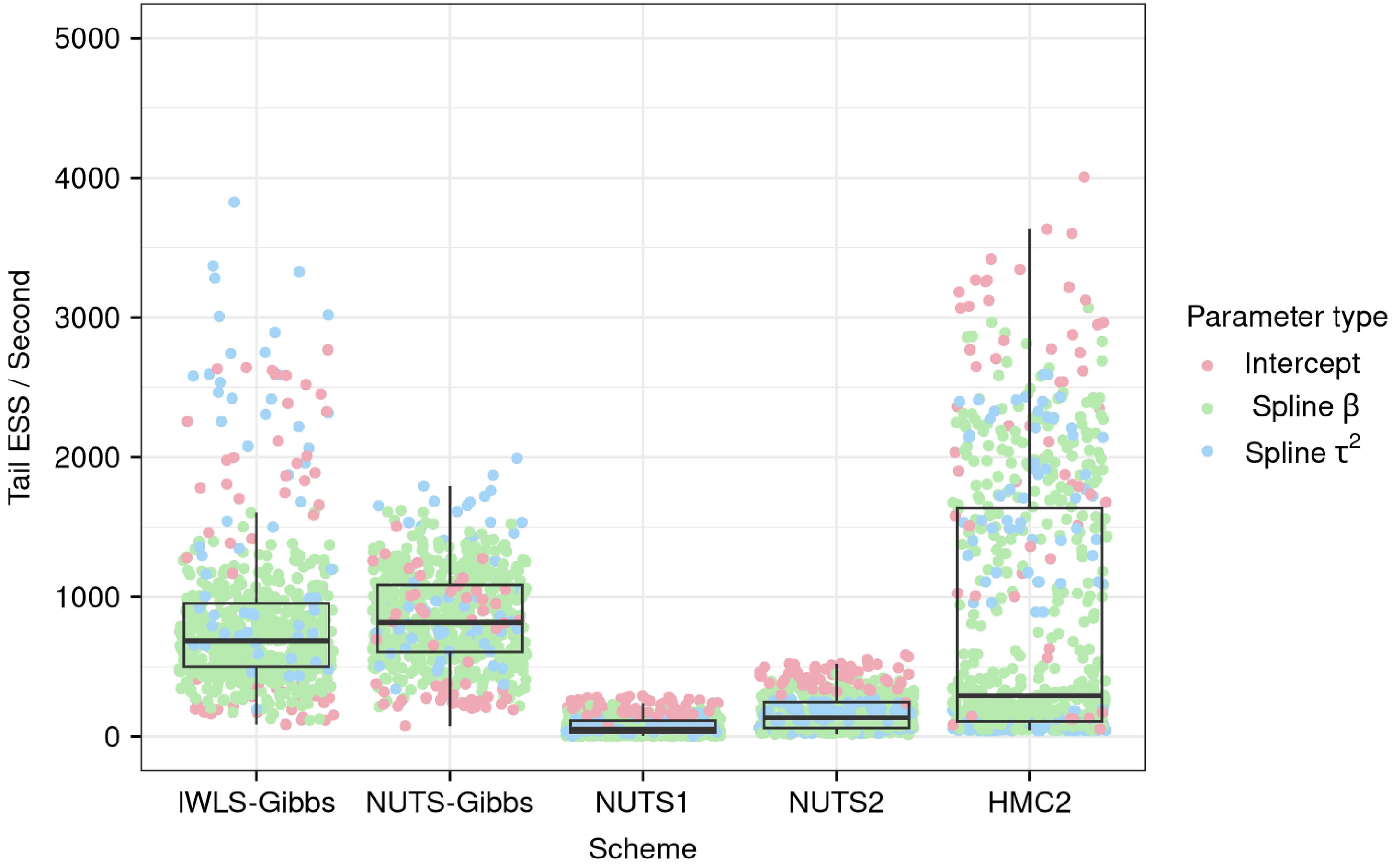
<table>
<tr><th></th><th>β_0</th><th>$\boldsymbol{\beta}_f$</th><th>τ_f^2 or $\log(\tau_f^2)$</th><th>γ_0</th><th>$\boldsymbol{\gamma}_g$</th><th>τ_g^2 or $\log(\tau_g^2)$</th></tr>
<tr><td><b>IWLS-Gibbs</b></td><td>IWLS</td><td>IWLS</td><td>Gibbs</td><td>IWLS</td><td>IWLS</td><td>Gibbs</td></tr>
<tr><td><b>NUTS-Gibbs</b></td><td>NUTS</td><td>NUTS</td><td>Gibbs</td><td>NUTS</td><td>NUTS</td><td>Gibbs</td></tr>
<tr><td><b>NUTS1</b></td><td colspan="6">NUTS</td></tr>
<tr><td><b>NUTS2</b></td><td colspan="3">NUTS</td><td colspan="3">NUTS</td></tr>
<tr><td><b>HMC2</b></td><td colspan="3">HMC</td><td colspan="3">HMC</td></tr>
</table>

Configuration of the sampling schemes

Case study 3: comparing samplers (cont'd)



Case study 3: comparing samplers (cont'd)



Using Liesel in practice

Using Liesel with Conda

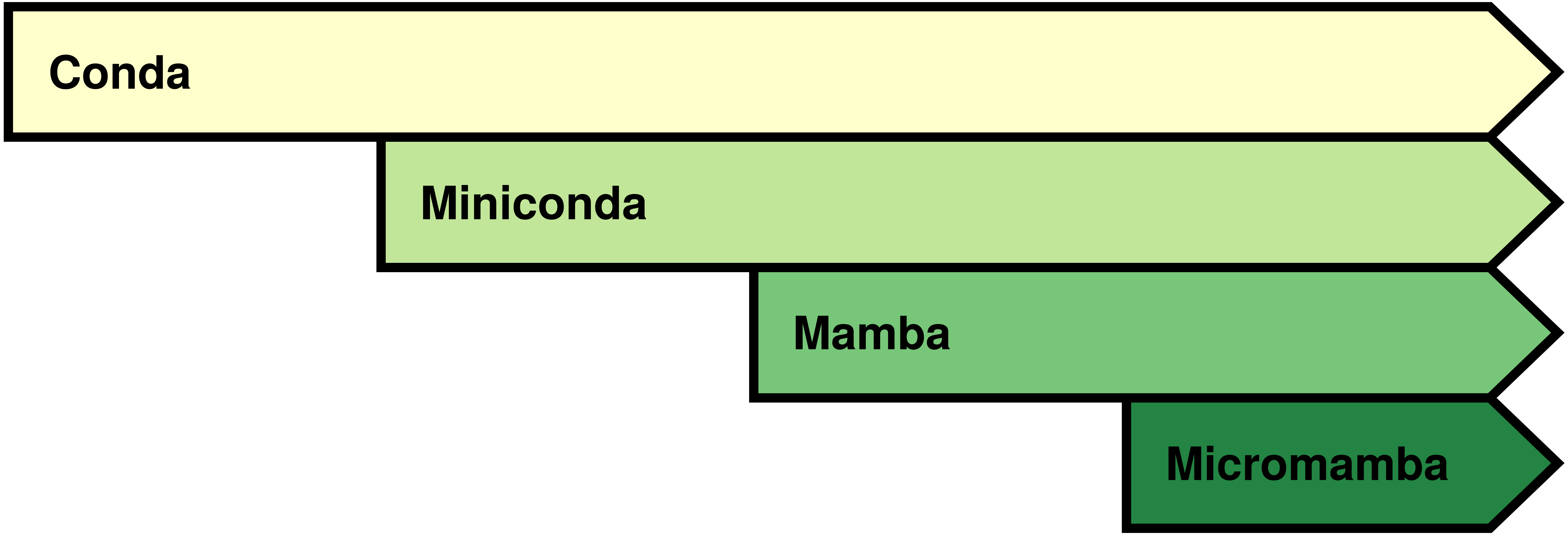
We are providing a template repository for Liesel-based projects: https://github.com/liesel-devs/liesel-template

Install Python, R, Liesel, RLiesel, Quarto and GNU Parallel with 5 commands:

```
git clone https://github.com/liesel-devs/liesel-template.git
cd liesel-template
micromamba env create -f environment.yml -p ./env
micromamba activate ./env
Rscript -e "remotes::install_github('liesel-devs/rliesel')"
```

With some caveats, of course… ¯_(ツ)_/¯

Using Liesel with Conda (cont’d)



Genealogy of Micromamba

Using Liesel with Conda (cont’d)

Packages managed by Micromamba:

```
(env) $ micromamba list
List of packages in environment: "env"

  Name                       Version       Build                Channel
____________________________________________________________________________
  _libgcc_mutex              0.1           conda_forge          conda-forge
  _openmp_mutex              4.5           2_gnu                conda-forge
  _r-mutex                   1.0.1         anacondar_1          conda-forge
  astunparse                 1.6.3         pyhd8ed1ab_0         conda-forge
  atk-1.0                    2.38.0        hd4edc92_1           conda-forge
  binutils_impl_linux-64     2.40          hf600244_0           conda-forge
  bwidget                    1.9.14        ha770c72_1           conda-forge
  bzip2                      1.0.8         hd590300_5           conda-forge
  c-ares                     1.23.0        hd590300_0           conda-forge
  ca-certificates            2023.11.17    hbcca054_0           conda-forge
  cairo                      1.18.0        h3faef2a_0           conda-forge
  cffi                       1.16.0        py311hb3a22ac_0      conda-forge
  cfgv                       3.3.1         pyhd8ed1ab_0         conda-forge
  colorama                   0.4.6         pyhd8ed1ab_0         conda-forge
  curl                       8.5.0         hca28451_0           conda-forge
```

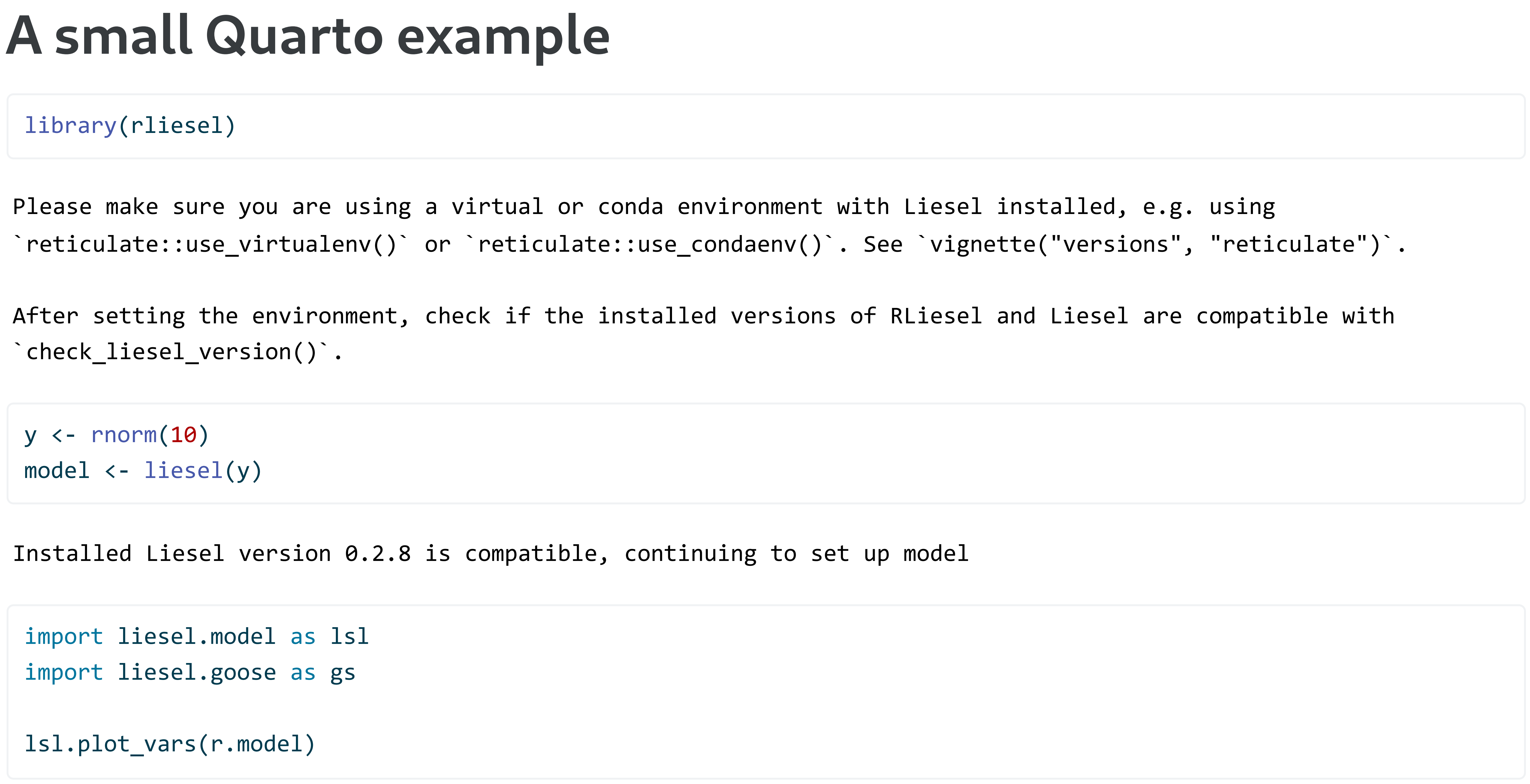
Using Liesel with Quarto

Combining RLiesel and Liesel in Quarto with R and Python code cells:

~~~
# A small Quarto example

```{r}
library(rliesel)

y <- rnorm(10)
model <- liesel(y)
```

```{python}
import liesel.model as lsl
import liesel.goose as gs

lsl.plot_vars(r.model)

builder = lsl.dist_reg_mcmc(r.model, seed=1337, num_chains=4)
builder.set_duration(warmup_duration=1000, posterior_duration=1000)
engine = builder.build()
engine.sample_all_epochs()

~~~

Using Liesel with Quarto (cont'd)

`quarto render quarto-example.qmd` compiles this to:



A small Quarto example

```r
library(rliesel)
```

```
Please make sure you are using a virtual or conda environment with Liesel installed, e.g. using
`reticulate::use_virtualenv()` or `reticulate::use_condaenv()`. See `vignette("versions", "reticulate")`.

After setting the environment, check if the installed versions of RLiesel and Liesel are compatible with
`check_liesel_version()`.
```

```r
y <- rnorm(10)
model <- liesel(y)
```

```
Installed Liesel version 0.2.8 is compatible, continuing to set up model
```

```python
import liesel.model as lsl
import liesel.goose as gs

lsl.plot_vars(r.model)
```

Using Liesel for simulation studies

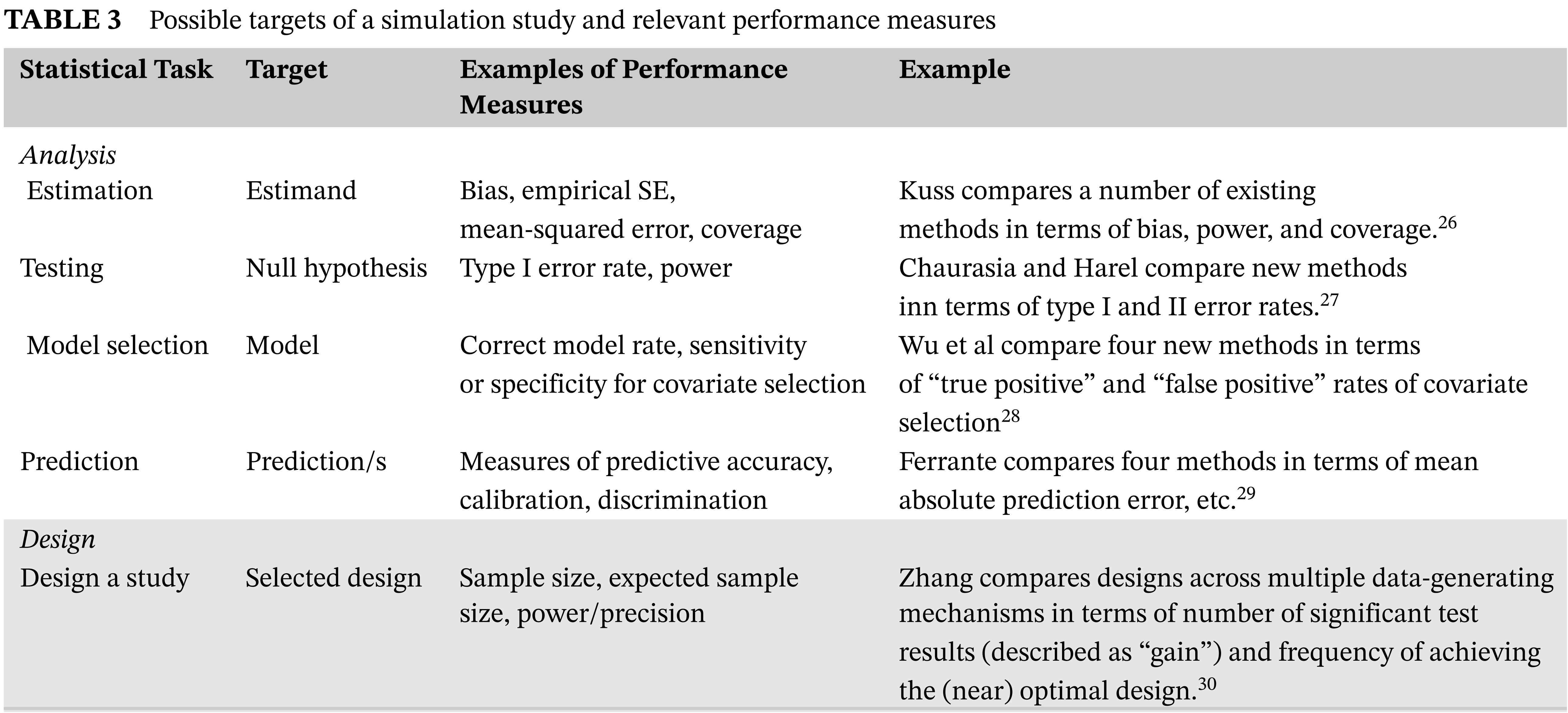


TABLE 3 Possible targets of a simulation study and relevant performance measures

Statistical Task	Target	Examples of Performance Measures	Example
Analysis			
Estimation	Estimand	Bias, empirical SE, mean-squared error, coverage	Kuss compares a number of existing methods in terms of bias, power, and coverage.[26]
Testing	Null hypothesis	Type I error rate, power	Chaurasia and Harel compare new methods inn terms of type I and II error rates.[27]
Model selection	Model	Correct model rate, sensitivity or specificity for covariate selection	Wu et al compare four new methods in terms of “true positive” and “false positive” rates of covariate selection[28]
Prediction	Prediction/s	Measures of predictive accuracy, calibration, discrimination	Ferrante compares four methods in terms of mean absolute prediction error, etc.[29]
Design			
Design a study	Selected design	Sample size, expected sample size, power/precision	Zhang compares designs across multiple data-generating mechanisms in terms of number of significant test results (described as “gain”) and frequency of achieving the (near) optimal design.[30]

Possible aims of simulation studies
(from Morris et. al., Using simulation studies to evaluate statistical methods)

Using Liesel for simulation studies (cont’d)

~~~
---
params:
  seed: 42
---

```{r}
set.seed(params$seed)
sample(10, 1)
```
~~~

`quarto render quarto-params.qmd` compiles this to:

```r
set.seed(params$seed)
sample(10, 1)
```

```
[1] 1
```

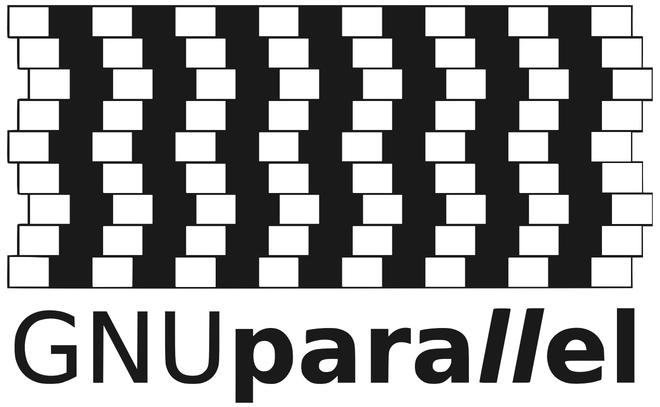
Using Liesel for simulation studies (cont’d)

`quarto render quarto-params.qmd -P seed:1337` compiles this to:

```r
set.seed(params$seed)
sample(10, 1)
```

```
[1] 10
```

Using Liesel for simulation studies (cont'd)



We have good experience with GNU Parallel for simulation studies with Liesel.

One brief example:

```
NREPS=4

parallel --jobs 4 --keep-order --plus --progress "
    mkdir -p jobs/job-{0#} &&
    cp scenario.qmd jobs/job-{0#} &&
    quarto render jobs/job-{0#}/scenario.qmd -P job:{#} &&
    rm jobs/job-{0#}/scenario.qmd" ::: $(seq $NREPS)
```

Options for the remaining time

1. Finish the exercise sheets. We are here to help and answer questions.
2. Install and run Liesel and RLiesel using the template repository.
3. Which models would **you** like to build with Liesel for your own research?
   New response structures, covariate effects, prior hierarchies?
4. What use cases do **you** have for parallelized simulation studies?