Introduction and structure of *ipmr*

ipmr

init_ipm()

define_kernel()

define_impl()

define_domains()

define_pop_state()

make_ipm()

- Vital rate modelling left outside the package

- Designed to reflect mathematical notation

- Reduces required time and coding knowledge for building IPMs

ipmr structure

```r
my_ipm <- init_ipm(sim_gen = "simple",
                   di_dd = "di",
                   det_stoch = "det")
```

init_ipm()

define_kernel()

define_impl()

define_domains()

define_pop_state()

make_ipm()

ipmr structure

init_ipm()
define_kernel()
define_impl()
define_domains()
define_pop_state()
make_ipm()

```
my_ipm <- define_kernel(proto_ipm = my_ipm,
                        name      = "P",
                        family    = "CC",

                        formula   = s * g,
                        s         = plogis(s_int + s_slope * dbh_1),
                        g         = dnorm(dbh_2, g_mu, g_sd),
                        g_mu      = g_int + g_slope * dbh_1,

                        data_list = named_data_list,

                        states        = list(c('dbh')),
                        evict_cor     = TRUE,
                        evict_fun     = truncated_distributions("norm", "g")
)
```

ipmr structure

init_ipm()

define_kernel()

define_impl()

define_domains()

define_pop_state()

make_ipm()

Math formula	R formula	ipmr
$\mu_G = \alpha_G + \beta_G * z$	size_2 ~ size_1, family =gaussian()	mu_G = G_int + G_slope * z
$G(z', z) = f_G(z', \mu_G, \sigma_G)$	G = dnorm(z_2, mu_G, sd_G)	G = dnorm(z_2, mu_G, sd_G)
$\text{logit}(s(z)) = \alpha_s + \beta_s * z$	surv ~size_1, family = binomial()	s = plogis(s_int + s_slope * z)
$\log(r_n(z)) = \alpha_{r_n} + \beta_{r_n} * z$	fec ~size_1, family = poisson()	r_n = exp(r_n_int + r_n_slope * z)
$\text{logit}(r_p(z)) = \alpha_{r_p} + \beta_{r_p} * z$	repr ~ size_1, family = binomial()	r_p = plogis(r_p_int + r_p_slope * z)
$r_d(z') = f_{r_d}(z', \mu_{r_d}, \sigma_{r_d})$	dnorm(z_2, mu_f_d, sigma_f_d)	r_d = dnorm(z_2, f_d_mu, f_d_sigma)

Levin et al., 2021 – (part of) Table 1

ipmr structure

init_ipm()
define_kernel()
define_impl()
define_domains()
define_pop_state()
make_ipm()

```
my_ipm <- define_impl(
  proto_ipm = my_ipm,
  kernel_impl_list = list(
    P = list(int_rule   = "midpoint",
             state_start = "dbh",
             state_end   = "dbh"),
    F = list(int_rule   = "midpoint",
             state_start = "dbh",
             state_end   = "dbh")
  )
  )
```

ipmr structure

init_ipm()

define_kernel()

define_impl()

define_domains()

define_pop_state()

make_ipm()

```
my_ipm <- define_domains(my_ipm,
                         # c(L, U, n_meshpoints)
                         dbh = c(1, 30, 200)
)

my_ipm <- define_pop_state(my_ipm,
                           n_dbh = rep(1/200, 200))


my_ipm <- make_ipm(my_ipm,
                   iterations = 100)
```