A comparison of frequentist and Bayesian SITAR model fit

Satpal Sandhu

January 13, 2023

Contents

- 1 Introduction
- 2 Frequentist SITAR model
- 3 Bayesian SITAR model
- 4 Comparison of growth curves estimated by the frequentist and Bayesian SITAR models
- $5\,\,$ Posterior predictive check for the Bayesian SITAR model
- 6 Diagnostic (trace and density) plots for the Bayesian SITAR model
- 7 References

1 Introduction

The super imposition by translation and rotation (SITAR) model (T. J. Cole et al., 2010) is a shape-invariant nonlinear mixed effect growth curve model that fits a population average (i.e., mean average) curve to the data and aligns each individual's growth trajectory to the population average curve by a set of three random effects (size, timing, and intensity). The SITAR has been specifically developed to model the human physical growth such as increase in height and weight gain during adolescence.@ref()

There are two competing philosophies of model estimation (Bland & Altman, 1998; Schoot et al., 2014): the Bayesian (based on Bayes' theorem) and the frequentist (e.g., maximum likelihood estimation) approaches. The frequentist version of the SITAR model can be fit by an already available R package, sitar (T. Cole, 2022). The bsitar package we introduce here implements the Bayesian SITAR model. The package is a frontend to the R package brms (Bürkner, 2022), which facilitates a range of hierarchical linear and nonlinear regression specifications including univariate, univariate univariate-by-subgroup (by using the subset functionality) and the multivariate modeling. The brms uses the 'Stan' software (Gelman et al., 2015) for full Bayesian inference

Like brms, the **bsitar** package allows a wide range of prior specifications that encourage the users to apply prior distributions that actually reflect their prior knowledge. Models can easily be evaluated and compared using several methods assessing posterior or prior predictions. The excellent post-processing support offered by the brms is further extended to predict and plot distance and growth velocity, and to compute growth parameters such as age at peak growth velocity and the peak growth velocity. The purpose of this document is to compare the results for the frequentist and the Bayesian implementation of SITAR model by using the sitar and bsitar packages. The only difference between these two packages is that while the sitar uses the B splines for the natural cubic spline design matrix(by calling the ns function of the splines package (R Core Team, 2022)), the bsitar constructs the spline design matrix by using the truncated power basis function approach as described by (Harrell & others, 2001) and implemented in the respline.eval function of the Hmisc package (Harrell Jr, 2022). Note that the **bsitar** package does not use this respline.eval but rather a custom function is written that is included in the Stan programs' functions block and thus compiled via the c++.

2 Frequentist SITAR model

As mentioned above Introduction, the *bsitar* package uses the truncated power basis function to construct the spline design matrix whereas the the *sitar* package implements the B spline approach. Both these approach results in the natural cubic spline characterized by continous first and second derivatives at knots, and a linear trajectory beyond the boundry knots, i.e., before the first and after the last knots (Harrell & others, 2001; R Core Team, 2022). However, to ascertain whether both these approach results in similar parameter estimates, below we first compare the results obtained from the *sitar* package (B spline basis) and its modified version to implement the truncated power basis approach.

We analyse the Berkeley growth data that are avaiable from the Berkeley Child Guidance Study conducted in Berkeley, California (Tuddenham & Snyder, 1954). Data contains longitudinal anthropometric measurements (such as height, weight etc.) made on 136 children (66 boys and 70 girls) born in 1928-29 of north European ancestry. The children were followed from birth to 21 years. Measurements were at ages 0, 0.085, 0.25 to 2 (3-monthly), 2 to 8 (annually), and 8 to 21 (6-monthly) years. We select a subset of data for male with height measurements obtained between 6 and 20 years of age.

Fit SITAR model (B splines - ns function from splines package i.e., default in the sitar package)

```
library(sitar) # Load 'sitar' package

# Fit model
sitar_ns <- sitar::sitar(x = age, y = height, id = id, data = data_male, df = 4)</pre>
```

Fit SITAR model (truncated power basis - sitar package modified for the respline.eval function from the Hmisc package)

```
# First, get knots from the above fitted sitar_ns model
knots_ns <- attr(sitar_ns$ns$model$ns, 'knots')
bounds_ns <- attr(sitar_ns$ns$model$ns, 'Boundary.knots')
knots <- sort(c(rep(bounds_ns), knots_ns)) + sitar_ns$xoffset

# Load sitar package modified for the truncated power basis
function_dir <- "E:/wdRfunctions/sitar_myfun"
sitar_function_load <- 'sitar_rcs_function_2022.R'
source(file.path(function_dir, sitar_function_load))

# Fit model
sitar_rcspline <- sitar(x = age, y = height, id = id, data = data_male, knots = knots)</pre>
```

Results for frequentist SITAR models estimated using the B spline basis (Table 1) and the truncated power basis function (see 2) are shown below. These results are summarized using the $nlme_tidiers$ function of the broo.mixed package (by setting the class ob object to 'lme') with default settings (conf.level = 0.95). The broo.mixed does not report CIs for the residual standard deviation. Findings show that except for the spline coefficients (i.e, s1, s2 etc.), both these approach results in identical estimates. These findings support a recent study (Sandhu, 2020) that used the truncated power basis approach to fit the SITAR model by using a modified sitar package.

Table 1: B spline basis ('sitar' package)

Parameter	Term	Estimate	Lower CI	Upper CI
Fixed	a	128.11	126.23	129.99
Fixed	b	-0.18	-0.40	0.04
Fixed	\mathbf{c}	0.41	0.36	0.45
Fixed	s1	25.71	24.85	26.57
Fixed	s2	44.37	43.25	45.48
Fixed	s3	64.45	62.68	66.21
Fixed	s4	42.27	41.35	43.20
Random (SD)	sd_a	6.18	5.21	7.34
Random (SD)	sd_b	0.87	0.73	1.03
Random (SD)	sd_c	0.13	0.11	0.16
Random (Corr)	$cor_b.a$	0.16	-0.08	0.38
Random (Corr)	cor_c.a	0.38	0.16	0.57
Random (Corr)	$cor_c.b$	-0.71	-0.81	-0.57
Residual (SD)	sd	0.77	NA	NA

Table 2: Truncated power basis (modified 'sitar' package)

Parameter	Term	Estimate	Lower CI	Upper CI
Fixed	a	128.11	126.23	129.99
Fixed	b	-0.18	-0.40	0.04
Fixed	\mathbf{c}	0.41	0.36	0.45
Fixed	s1	4.00	3.86	4.15
Fixed	s2	-2.87	-3.17	-2.57
Fixed	s3	24.54	23.10	25.97
Fixed	s4	-60.67	-62.98	-58.37
Random (SD)	sd_a	6.18	5.21	7.34
Random (SD)	sd_b	0.87	0.73	1.03
Random (SD)	sd_c	0.13	0.11	0.16
Random (Corr)	$cor_b.a$	0.16	-0.08	0.38
Random (Corr)	$cor_c.a$	0.38	0.16	0.57
Random (Corr)	$cor_c.b$	-0.71	-0.81	-0.57
Residual (SD)	sd	0.77	NA	NA

3 Bayesian SITAR model

We fit the Bayesian SITAR model to the same data as described above (see Frequentist SITAR model section for details). Full syntax for the *bsitar* function showing the deafult prior and initial values for the population average parameters and the individual level random effects parameters (standard deviation and correlation) is provided in Appendix A.

A detailed output is shown below followed by a summary of estimates as Table 3. Results are summarized using the *brms_tidiers* function of the *broo.mixed* package with default settings (conf.level = 0.95). Results are similar to the those reported earlier for the frequntist SITAR model (see Table 1 and Table 2). The spline coefficients are almost identical to those obtained from the frequntist SITAR model with truncated power basis approach (Table 2). The other parameters including population average estimates and the variance covariance estimate are also similar (though the standard deviation for the size parameter (a) is slightly higher for the Bayesian model than the frequntist model).

```
library(bsitar) # Load 'bsitar' package

# Fit model with default settings (see Appendix A for full options)
    # 4 chains with 2000 iterrations per chain (1000 warmup)
    # backend = 'rstan'

# Fit model - Note the exact same syntax as sitar::sitar

bsitar_b <- bsitar::bsitar(x = age, y = height, id = id, data = data_male, df = 4)</pre>
```

```
# Summary of Bayesian SITAR model
print(bsitar_b)
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: height ~ SplineFun(age, a, b, c, s1, s2, s3, s4)
         a ~ 1 + (1 | C | gr(id, dist = "gaussian"))
         b ~ 1 + (1 | C | gr(id, dist = "gaussian"))
         c ~ 1 + (1 | C | gr(id, dist = "gaussian"))
         s1 ~ 1
         s2 ~ 1
         s3 ~ 1
         s4 \sim 1
  Data: structure(list(id = structure(c(1L, 1L, 1L, 1L, 1L, 1L (Number of observations: 1623)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000
Group-Level Effects:
~id (Number of levels: 66)
                             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS
sd(a_Intercept)
                                  6.40
                                            0.59
                                                     5.32
                                                              7.72 1.01
                                                                              322
sd(b_Intercept)
                                  0.91
                                            0.08
                                                     0.77
                                                              1.08 1.01
                                                                              372
sd(c_Intercept)
                                  0.13
                                            0.01
                                                     0.11
                                                              0.16 1.01
                                                                              436
cor(a_Intercept,b_Intercept)
                                                    -0.06
                                                                              293
                                 0.18
                                            0.12
                                                              0.39 1.01
cor(a_Intercept,c_Intercept)
                                  0.36
                                            0.11
                                                     0.14
                                                              0.56 1.01
                                                                              322
cor(b_Intercept,c_Intercept)
                                -0.70
                                            0.06
                                                    -0.81
                                                             -0.56 1.01
                                                                              417
                             Tail ESS
sd(a_Intercept)
                                   479
sd(b_Intercept)
                                   797
sd(c Intercept)
                                   689
cor(a_Intercept,b_Intercept)
                                   567
cor(a_Intercept,c_Intercept)
                                   641
cor(b_Intercept,c_Intercept)
                                   633
Population-Level Effects:
             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
a_Intercept
               128.10
                           0.99
                                  126.23
                                            130.06 1.01
                                                             298
                                                                       629
b_Intercept
                -0.17
                           0.12
                                   -0.39
                                              0.07 1.01
                                                             246
                                                                       298
                 0.41
                           0.02
                                    0.36
                                              0.45 1.01
                                                             358
c_Intercept
                                                                       876
s1_Intercept
                 4.00
                           0.08
                                    3.84
                                              4.16 1.00
                                                            1223
                                                                      1529
                -2.84
                                             -2.51 1.00
s2_Intercept
                           0.16
                                   -3.17
                                                            1065
                                                                      1548
s3 Intercept
                24.39
                           0.78
                                    22.84
                                             25.98 1.00
                                                            1093
                                                                      1526
               -60.44
                           1.26
                                   -62.92
                                            -57.93 1.00
                                                            1065
                                                                      1457
s4_Intercept
Family Specific Parameters:
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
          0.78
                    0.01
                             0.75
                                       0.81 1.00
                                                     1967
sigma
Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Table 3: Bayesian SITAR model

Parameter	Term	Estimate	Lower CI	Upper CI
Fixed	a	128.10	126.24	130.06
Fixed	b	-0.17	-0.39	0.07
Fixed	\mathbf{c}	0.41	0.36	0.45
Fixed	s1	4.00	3.84	4.16
Fixed	s2	-2.84	-3.17	-2.51
Fixed	s3	24.39	22.84	25.98
Fixed	s4	-60.44	-62.92	-57.93
Random (SD)	sd_a	6.40	5.32	7.72
Random (SD)	sd_b	0.91	0.77	1.08
Random (SD)	sd_c	0.13	0.11	0.16
Random (Corr)	cor_a_b	0.18	-0.06	0.39
Random (Corr)	cor_a_c	0.36	0.14	0.56
Random (Corr)	cor_b_c	-0.70	-0.81	-0.56
Residual (SD)	sd	0.78	0.75	0.81

4 Comparison of growth curves estimated by the frequentist and Bayesian SITAR models

Here we show comparison of distance and velocity curves estimated by the **bsitar** and the **sitar** with the log and square root transformation of predictor (age) and the outcome.

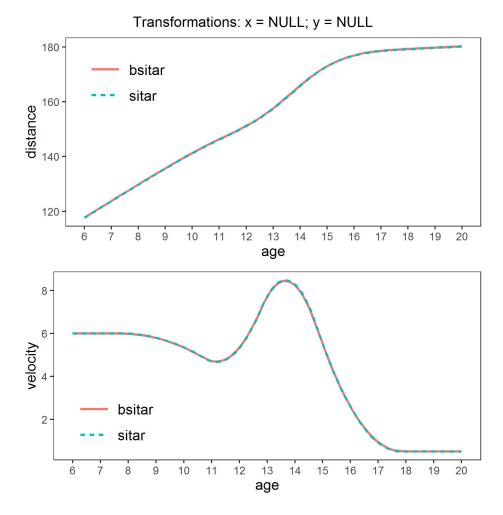


Figure 1: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with original age and outcome

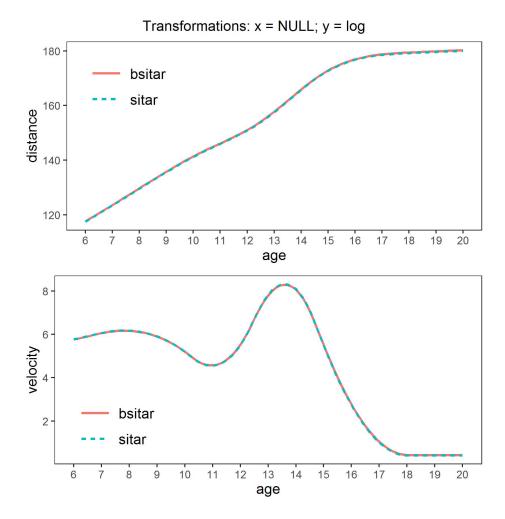


Figure 2: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with original age and log transformed outcome

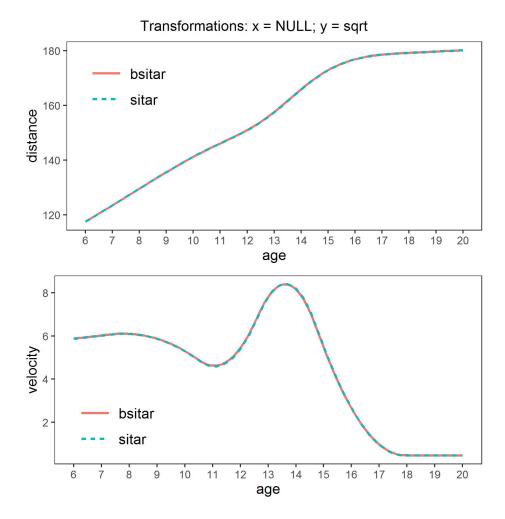


Figure 3: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with original age and square root transformed outcome

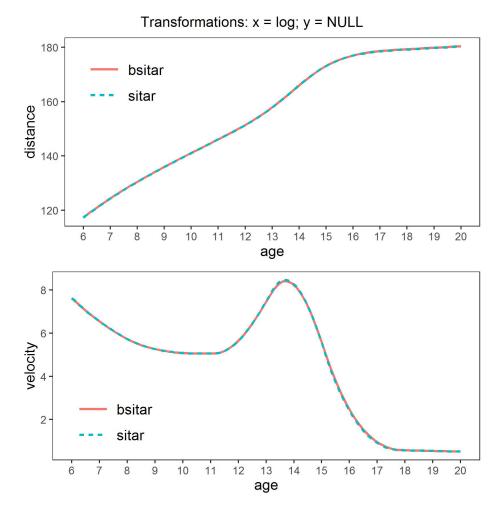


Figure 4: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with log transformed age and original outcome

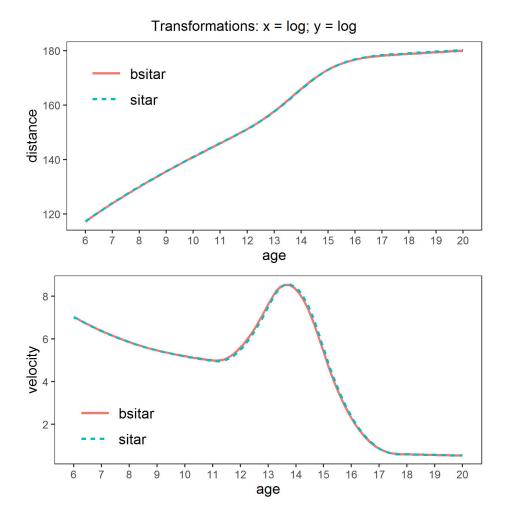


Figure 5: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with log transformed both age and outcome

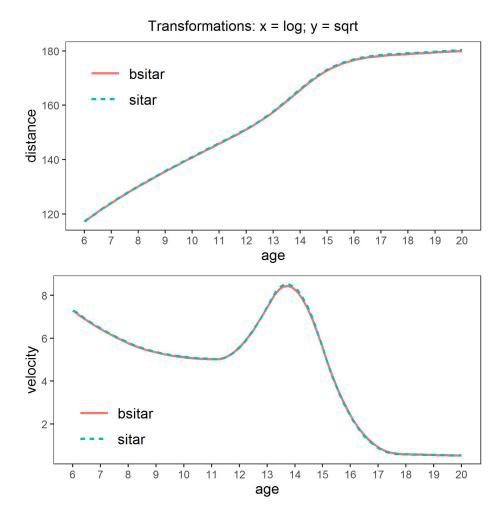


Figure 6: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with log transformed age and square root transformed outcome

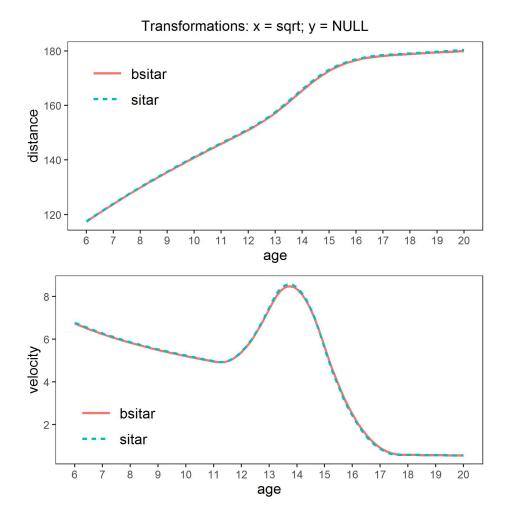


Figure 7: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with square root transformed age and original outcome

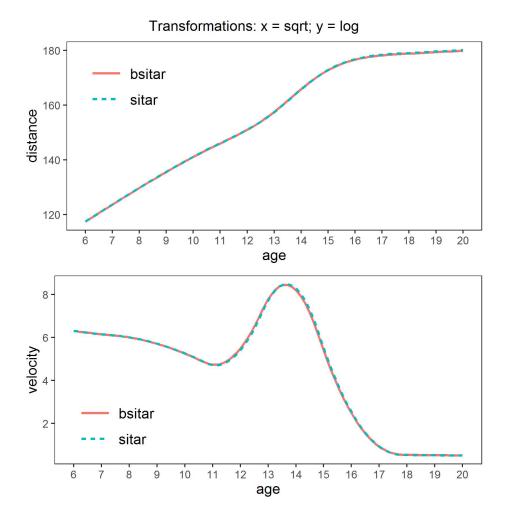


Figure 8: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with square root transformed age and log transformed outcome

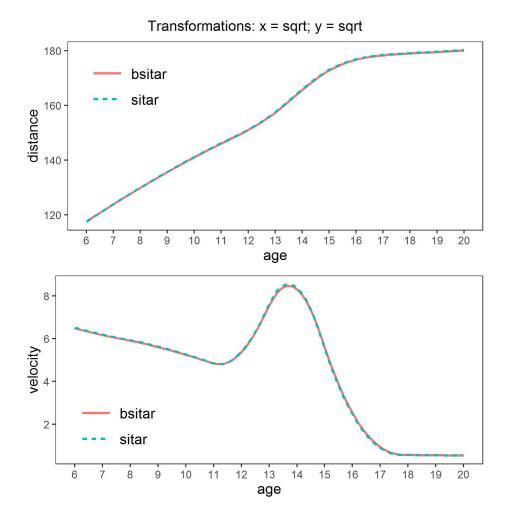


Figure 9: Comparison of distance and velocity curves estimated by the frequntist and Bayesian SITAR models applied to the data with square root transformed both age and outcome

5 Posterior predictive check for the Bayesian SITAR model

This sections shows the posterior predictive checks (pp checks) for the Bayesian SITAR model estimated using the **bsitar** package. The SITAR model are estimated with and without the log and square root transformation of predictor (age) and the outcome.

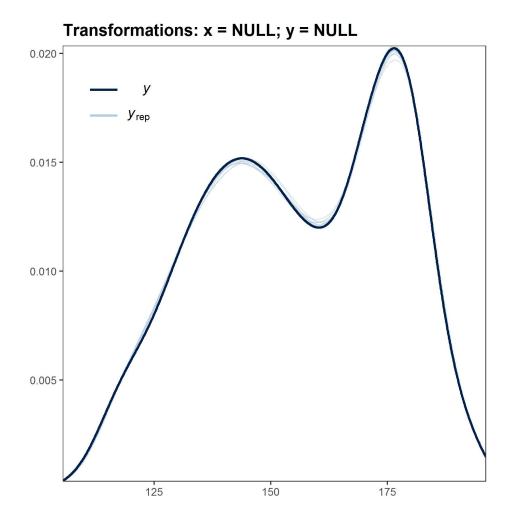


Figure 10: Posterior predictive check for the Bayesian SITAR models applied to the data with original age and outcome

Transformations: x = NULL; y = log

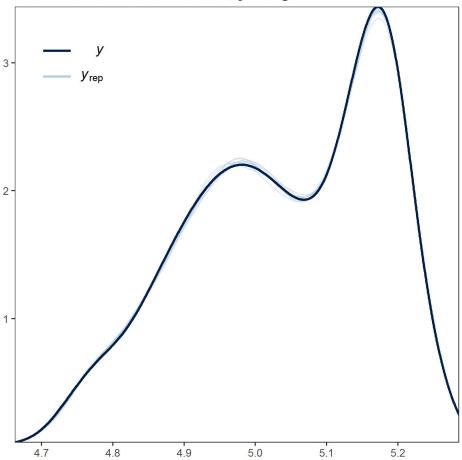


Figure 11: Posterior predictive check for the Bayesian SITAR models applied to the data with original age and \log transformed outcome

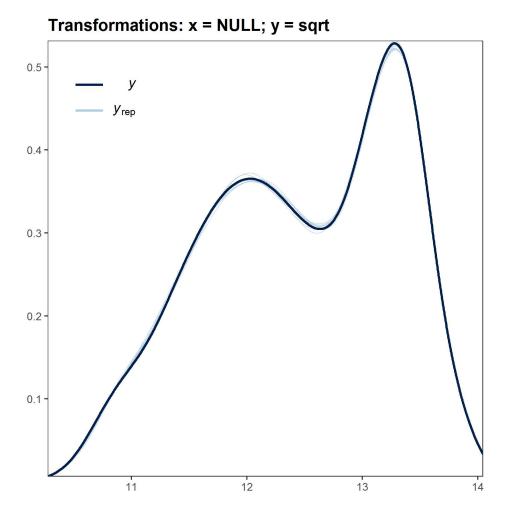


Figure 12: Posterior predictive check for the Bayesian SITAR models applied to the data with original age and square root transformed outcome

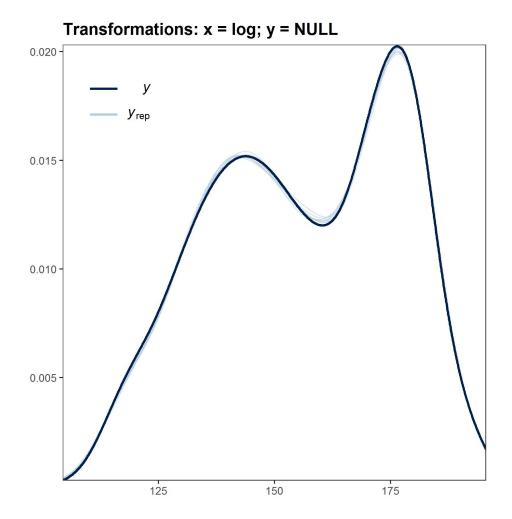


Figure 13: Posterior predictive check for the Bayesian SITAR models applied to the data with \log transformed age and original outcome

Transformations: x = log; y = log

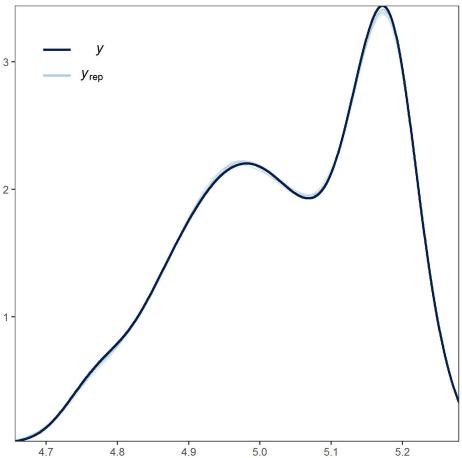


Figure 14: Posterior predictive check for the Bayesian SITAR models applied to the data with log transformed both age and outcome

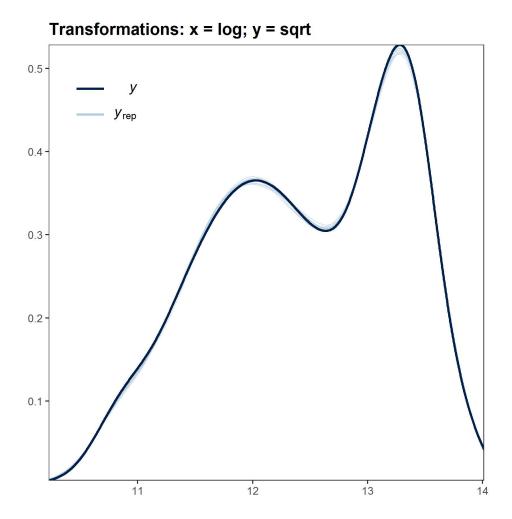


Figure 15: Posterior predictive check for the Bayesian SITAR models applied to the data with \log transformed age and square root transformed outcome

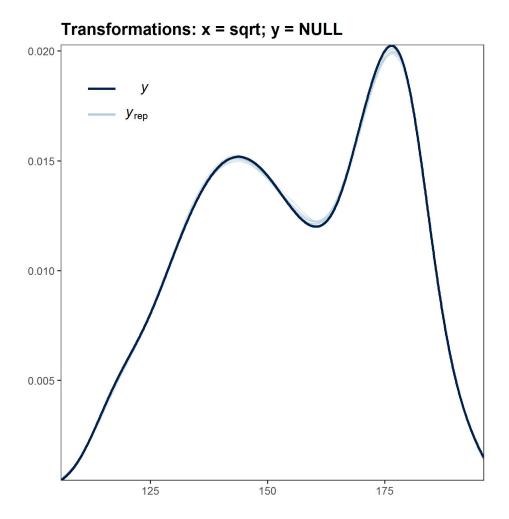


Figure 16: Posterior predictive check for the Bayesian SITAR models applied to the data with square root transformed age and original outcome

Transformations: x = sqrt; y = log

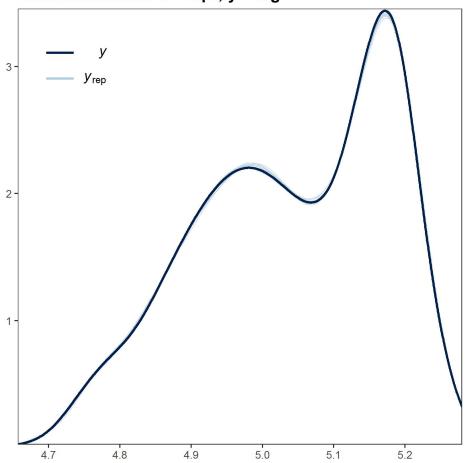


Figure 17: Posterior predictive check for the Bayesian SITAR models applied to the data with square root transformed age and log transformed outcome

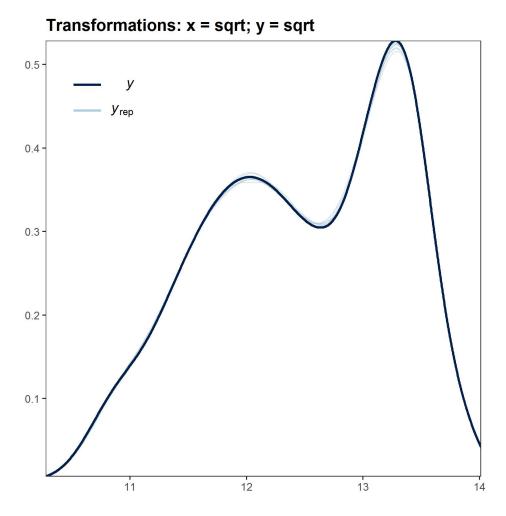
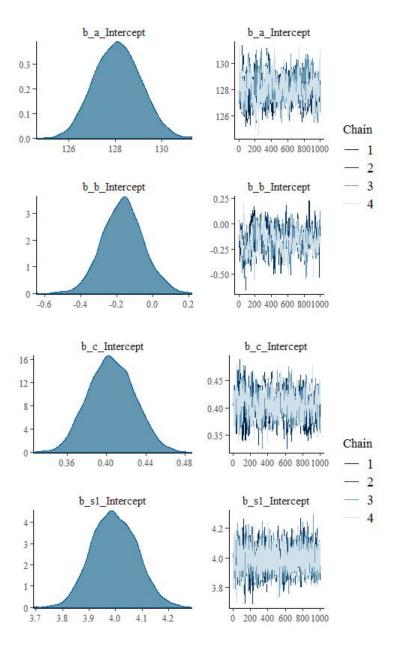
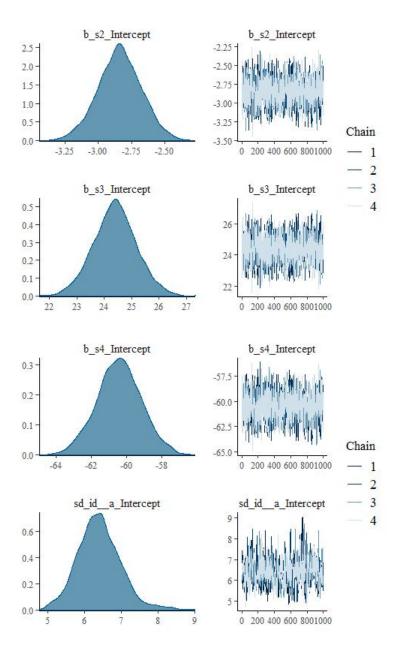
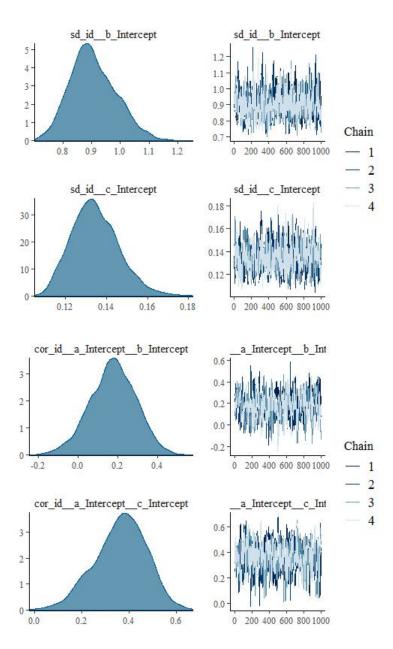


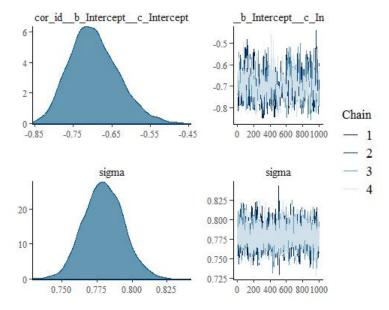
Figure 18: Posterior predictive check for the Bayesian SITAR models applied to the data with square root transformed both age and outcome

6 Diagnostic (trace and density) plots for the Bayesian SITAR model









7 References

- Bland, J. M., & Altman, D. G. (1998). Statistics notes: Bayesians and frequentists. *BMJ*: *British Medical Journal*, 317(7166), 1151. https://doi.org/10.1136/bmj.317.7166.1151
- Bürkner, P.-C. (2022). Brms: Bayesian regression models using stan. https://CRAN.R-project.org/package=brms
- Cole, T. (2022). Sitar: Super imposition by translation and rotation growth curve analysis. https://github.com/statist7/sitar
- Cole, T. J., Donaldson, M. D. C., & Ben-Shlomo, Y. (2010). SITAR—a useful instrument for growth curve analysis. *International Journal of Epidemiology*, 39(6), 1558–1566. https://doi.org/10.1093/ije/dyq115
- Gelman, A., Lee, D., & Guo, J. (2015). Stan: A Probabilistic Programming Language for Bayesian Inference and Optimization. *Journal of Educational and Behavioral Statistics*, 40(5), 530–543. https://doi.org/10.3102/1076998615606113
- Harrell, F. E., & others. (2001). Regression modeling strategies: With applications to linear models, logistic regression, and survival analysis (Vol. 608). Springer.
- Harrell Jr, F. E. (2022). Hmisc: Harrell miscellaneous. https://CRAN.R-project.org/package=Hmisc
- R Core Team. (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing. https://www.R-project.org/
- Sandhu, S. S. (2020). Analysis of longitudinal jaw growth data to study sex differences in timing and intensity of the adolescent growth spurt for normal growth and skeletal discrepancies [Thesis].
- Schoot, R. van de, Kaplan, D., Denissen, J., Asendorpf, J. B., Neyer, F. J., & Aken, M. A. G. van. (2014). A gentle introduction to bayesian analysis: Applications to developmental research [Journal Article]. *Child Dev*, 85(3), 842–860. https://doi.org/10.1111/cdev.12169
- Tuddenham, R. D., & Snyder, M. M. (1954). Physical growth of California boys and girls from birth to eighteen years. *Publications in Child Development. University of California, Berkeley*, 1(2), 183–364.

```
# Full syntax of 'bsitar' function to fit Bayesian SITAR model
bsitar <- function(x,</pre>
                   у,
                   id,
                   data,
                   df = 4,
                   knots = NA,
                   fixed = a + b + c,
                   random = a + b + c,
                   xoffset = mean,
                   bstart = mean,
                   xfun = NULL,
                   yfun = NULL,
                   bound = 0.04,
                   a_formula = ~ 1,
                   b_formula = ~ 1,
                   c_formula = ~ 1,
                   d_formula = ~ 1,
                   s_formula = ~1,
                   a_formula_gr = ~ 1,
                   b_formula_gr = ~ 1,
                   c_formula_gr = ~ 1,
                   d_formula_gr = ~ 1,
                   dpar formula = NULL,
                   autocor_formula = NULL,
                   family = gaussian(),
                   group_arg = list(
                     groupvar = NULL,
                     by = NULL,
                     cor = un,
                     cov = NULL,
                     dist = gaussian
                   univariate_by = list(by = NA, cor = un),
                   multivariate = list(mvar = FALSE,
                                        cor = un,
                                        rescor = TRUE),
                   a_prior_beta = normal(ymean, ysd, autoscale = 2.5),
                   b_prior_beta = normal(0, 2.5, autoscale = FALSE),
                   c_prior_beta = normal(0, 1, autoscale = FALSE),
                   d_prior_beta = normal(0, 1, autoscale = FALSE),
                   s_prior_beta = normal(0, lm, autoscale = 2.5),
                   a_cov_prior_beta = normal(0, 10, autoscale = FALSE),
                   b_cov_prior_beta = normal(0, 2, autoscale = FALSE),
                   c_cov_prior_beta = normal(0, 0.1, autoscale = FALSE),
                   d_cov_prior_beta = normal(0, 1, autoscale = FALSE),
                   s_cov_prior_beta = normal(0, 10, autoscale = FALSE),
                   a_prior_sd = normal(0, ysd, autoscale = 1),
                   b_prior_sd = normal(0, 1.5, autoscale = FALSE),
                   c_prior_sd = normal(0, 0.5, autoscale = FALSE),
```

```
d_prior_sd = normal(0, 1, autoscale = FALSE),
a_cov_prior_sd = normal(0, 10, autoscale = FALSE),
b_cov_prior_sd = normal(0, 2, autoscale = FALSE),
c_cov_prior_sd = normal(0, 0.25, autoscale = FALSE),
d_cov_prior_sd = normal(0, 0.25, autoscale = FALSE),
gr_prior_cor = lkj(1),
rsd_prior_sigma = normal(0, ysd, autoscale = FALSE),
dpar_prior_sigma = normal(0, ysd, autoscale = FALSE),
dpar_cov_prior_sigma = normal(0, 5, autoscale = FALSE),
autocor_prior_acor = uniform(-1, 1, autoscale = FALSE),
mvr_prior_rescor = lkj(1),
init = NULL,
init_r = NULL,
a_init_beta = lm,
b_init_beta = 0.001,
c_init_beta = 0.001,
d_init_beta = 0.001,
s_init_beta = lm,
a_cov_init_beta = lm,
b_cov_init_beta = lm,
c_cov_init_beta = lm,
d_cov_init_beta = lm,
s_cov_init_beta = lm,
a_init_sd = lme_sd_a,
b_{init\_sd} = 0.1,
c_{init_sd} = 0.01,
d_{init\_sd} = 0.1,
a_cov_init_sd = 0,
b_cov_init_sd = 0,
c_cov_init_sd = 0,
d_cov_init_sd = 0,
gr_init_cor = 0,
rsd_init_sigma = 0.01,
dpar_init_sigma = 0,
dpar_cov_init_sigma = 0,
autocor_init_acor = 0.5,
mvr_init_rescor = 0,
r_{init_z} = 0,
jitter_init_beta = 0.01,
jitter_init_sd = NULL,
jitter_init_cor = NULL,
prior_data = NULL,
init_data = NULL,
verbose = FALSE,
expose_function = TRUE,
chains = 4.
iter = 2000,
warmup = floor(iter / 2),
thin = 1,
cores = getOption("mc.cores", "optimize"),
backend = getOption("brms.backend", "rstan"),
threads = getOption("brms.threads", "optimize"),
```

```
opencl = getOption("brms.opencl", NULL),
normalize = getOption("brms.normalize", TRUE),
algorithm = getOption("brms.algorithm", "sampling"),
control = list(adapt_delta = 0.8, max_treedepth = 15),
sample_prior = "no",
save_pars = NULL,
save_ranef = NULL,
save_mevars = NULL,
save_all_pars = NULL,
drop_unused_levels = TRUE,
stan_model_args = list(),
empty = FALSE,
rename = TRUE,
silent = 1,
seed = 123,
save_model = NULL,
fit = NA,
file = NULL,
file_refit = getOption("brms.file_refit", "never"),
future = getOption("future", FALSE),
```