Supplementary sections for "shinybrms: Fitting Bayesian Regression Models Using a Graphical User Interface for the R Package brms"

by Frank Weber, Katja Ickstadt, and Änne Glass

Existing GUIs

By the term "GUI", we denote here a software interface not requiring the knowledge of any command-line syntax (which would hinder its ease of use). A graphical interface for a command-line tool such as BayesX (Brezger et al., 2005; Belitz et al., 2015) is therefore not considered to be a GUI here.

Furthermore, there are GUIs designed only for specific purposes, e.g., the R package **beanz** (Wang et al., 2018) which uses Stan and provides a **shiny** (Chang et al., 2022) app for estimating the heterogeneity of a treatment effect between individuals. Even if these GUIs offer special regression models (as does **beanz**), here we only consider GUIs offering general-purpose regression models.

In Table 1 of the main article as well as in the following, we tried to order the existing GUIs for Bayesian regression models (BRMs) chronologically with respect to their first release (or—for general statistical software packages—with respect to their first release which included BRMs). As the exact date of the first release (or the first release which included BRMs) was not always available, we hope our ordering reflects the true chronological order as closely as possible.

The algorithms for inferring the posterior (as well as their abbreviations) are introduced in section "Algorithms for inferring the posterior" of the main article.

BUGS and JAGS

The first general-purpose language and software package for Bayesian modeling was BUGS (Gilks et al., 1994) which stands for "Bayesian inference using Gibbs sampling". The original BUGS software package only offers a command-line interface. Later, WinBUGS (Lunn et al., 2000) was released for Windows systems. WinBUGS allows to specify a model graphically *via* a *Doodle*. However, the principal way to specify models in WinBUGS is still *via* code. With OpenBUGS (Spiegelhalter et al., 2014), a platform-independent alternative to WinBUGS became available. JAGS (Plummer, 2003) (which stands for "Just another Gibbs sampler") may also be considered as a descendant of BUGS, but it offers no GUI at all. However, BugsXLA (Woodward, 2011) is a Microsoft Excel (Microsoft Corporation, 2016) add-in based on WinBUGS, OpenBUGS, and JAGS which offers a GUI for specific models (most importantly, BRMs). Therefore, with BugsXLA, we consider WinBUGS, OpenBUGS, and JAGS to have a full GUI for BRMs. We note that the R package **iBUGS** (Xie and Wang, 2013) also offers a GUI for WinBUGS, OpenBUGS, and JAGS, but it seems to be discontinued, so we do not consider it further here.

As suggested by their names, BUGS and all of its descendants use a Gibbs sampler structure. If needed, various other MC and MCMC algorithms are used for sampling from the full conditionals within this Gibbs sampler structure (Lunn et al., 2012; Congdon, 2020; Plummer, 2017). For OpenBUGS, this includes static HMC, but not the NUTS (Lunn et al., 2012).

BUGS and its descendants are noncommercial. An overview of the history of the BUGS project may be found in Lunn et al. (2009) as well as in shorter form in Plummer (2003).

IBM SPSS Amos

IBM SPSS Amos (Arbuckle, 2020) is a commercial software package with a GUI designed for Bayesian structural equation models, so it may also be used for BRMs. Algorithmically, IBM SPSS Amos relies on MCMC procedures (more precisely: the Metropolis algorithm and static HMC). IBM SPSS Amos also offers a syntax.

Toolkit on Econometrics and Economics Teaching

The MATLAB (The MathWorks, Inc., 2020a) package "Toolkit on Econometrics and Economics Teaching" (Qian, 2011) (here abbreviated by "TEET") includes some BRMs and offers a GUI. The TEET package itself is noncommercial, but MATLAB is commercial. TEET mainly uses MCMC algorithms (Gibbs sampling, MH-within-Gibbs, MH) but occasionally, TEET also uses other algorithms (no HMC, though), e.g., inverse transform sampling, custom analytic expressions, and expectation maximization with numerical integration.

JASP

JASP (JASP Team, 2022) is a noncommercial statistical software package with a GUI. At the time of writing, the most recent JASP version was 0.16.3. JASP offers a variety of frequentist and Bayesian methods, including (but not limited to) regression models.

According to the JASP source code (JASP Team, 2020b) and other online resources (JASP Team, 2020a,c), JASP uses different algorithms for inferring the posterior: analytic expressions, numerical integration, MC integration, JAGS (see Supplement section BUGS and JAGS), and a variety of external R packages which themselves use their own algorithms. It is *via* external R packages that JASP also uses Stan: Shortly after **shinybrms**'s (Weber, 2022) first release, JASP version 0.12 introduced a Bayesian meta-analysis feature based on Stan and JASP version 0.13 introduced mixed BRMs based on Stan.

Bayesian Regression: Nonparametric and Parametric Models

"Bayesian Regression: Nonparametric and Parametric Models" (Karabatsos, 2015, 2017) (here abbreviated by "BRNPM") is a noncommercial software package built with the MATLAB Compiler (The MathWorks, Inc., 2020b). It includes a variety of BRMs, in particular also infinite-mixture BRMs which may be seen as nonparametric alternatives to the commonly used BRMs. BRNPM also offers a variable selection *via* a spike-and-slab prior for the regression coefficients.

At its heart, the algorithm has a Gibbs sampler structure with various other MCMC algorithms (but not HMC) for sampling from the full conditionals within this Gibbs sampler structure (Karabatsos, 2017). For the nonparametric (infinite-mixture) BRMs, special handling is required (Karabatsos, 2017). Algorithmic details may be found in Karabatsos (2017).

Stata

Stata (StataCorp, 2019a) is a commercial statistical software package having a GUI as well as its own syntax. Bayesian methods were introduced in Stata version 14. Among these are BRMs with a variety of outcome distributions, priors, and other modeling features.

For the Bayesian methods, Stata relies on the following MCMC algorithms (StataCorp, 2019b):

- adaptive MH (possibly blocked),
- hybrid MH (which is a blocked adaptive MH algorithm with Gibbs updates in user-specified blocks),

- full Gibbs sampling (for some models).

The choice of the algorithm is mostly up to the user, with some exceptions explained in the "Stata Bayesian Analysis Reference Manual" (StataCorp, 2019b).

We note that there is also StataStan (Grant et al., 2017a,b), the Stata interface to Stan, but it does not feature a GUI.

BayES

BayES (Emvalomatis, 2020) is a software package for Bayesian analyses that was originally designed for application in econometrics. It includes some BRMs and uses direct Gibbs sampling whenever possible, otherwise MH-within-Gibbs (Emvalomatis, 2020). BayES offers a GUI as well as a syntax. The current BayES version 2.5 is free of charge.

IBM SPSS

IBM SPSS (IBM Corp., 2020) is a commercial statistical software package with a GUI as well as its own syntax. Support for Bayesian methods started with IBM SPSS version 25.0. However, IBM SPSS offers only a limited amount of Bayesian methods and only some of them are BRMs.

Computationally, IBM SPSS mainly seems to use analytic expressions. Numerical integration and integration by MC sampling are only offered for calculating Bayes factors, with a few exceptions where they are also used for inferring the posterior. The computational method for inferring the posterior is always chosen automatically, i.e., not by the user.

Generally, IBM SPSS places much importance on Bayes factors and decision thresholds for them, a hypothesis-testing approach which is discouraged (McShane et al., 2019).

BEsmarter

BEsmarter (BEsmarter Team, 2020a,b; Ramírez-Hassan and Graciano-Londoño, 2021) is a noncommercial **shiny** app originally developed in the context of econometrics. It includes a variety of BRMs and relies mostly on MCMC algorithms (MH, Gibbs sampling, and MH-within-Gibbs) (Ramírez-Hassan and Graciano-Londoño, 2021) but also offers the Bayesian bootstrap (Rubin, 1981) for linear regression models. Compared to the methods for inferring the posterior described in section "Algorithms for inferring the posterior" of the main article, the Bayesian bootstrap is a somewhat different approach as it requires "peculiar model assumptions" (Rubin, 1981).

Advanced distributional families

The following families are supported by **shinybrms** as "advanced" distributional families:

- Continuous outcome on the real line:
  - `brms::student()`
  - `brms::skew_normal()`
  - `brms::asym_laplace()`
- Continuous outcome on the positive (or nonnegative) real line:
  - `brms::lognormal()`
  - `brms::hurdle_lognormal()`
  - `stats::Gamma()`
  - `brms::hurdle_gamma()`

  - `stats::inverse.gaussian()`
  - `brms::weibull()`
  - `brms::exponential()`
  - `brms::frechet()`
- Count data outcome:
  - `brms::hurdle_negbinomial()`
  - `brms::zero_inflated_negbinomial()`
  - `stats::poisson()`
  - `brms::hurdle_poisson()`
  - `brms::zero_inflated_poisson()`
  - `brms::geometric()`
- Proportion as outcome:
  - `brms::Beta()`
  - `brms::zero_inflated_beta()`
  - `brms::zero_one_inflated_beta()`
- Circular outcome:
  - `brms::von_mises()`
- Response time outcome:
  - `brms::shifted_lognormal()`
  - `brms::exgaussian()`

Frequentist analysis of the example

In a frequentist context, an analogous regression model to the Bayesian one from section "Example" of the main article may be fitted using **lme4** (Bates et al., 2015):

```
> CAP <- read.csv("CAP.csv")
> library(lme4)
> lmm <- lmer(TWI ~ age + anticoagulation + diabetes + day * trt + (1 | patID),
+             data = CAP)
```

The output corresponding to that from **shinybrms**'s tab "Default summary" (Figure 9 of the main article) is given by:

```
> print(summary(lmm),
+       correlation = FALSE,
+       ranef.comp = "Std.Dev.",
+       show.resids = FALSE)
Linear mixed model fit by REML ['lmerMod']
Formula: TWI ~ age + anticoagulation + diabetes + day * trt + (1 | patID)
   Data: CAP

REML criterion at convergence: 2268.3

Random effects:
 Groups   Name        Std.Dev.
 patID    (Intercept) 3.172
 Residual             5.820
```

```
Number of obs: 360, groups:  patID, 10

Fixed effects:
                   Estimate Std. Error t value
(Intercept)         39.4827    11.0197   3.583
age                  0.1010     0.1574   0.641
anticoagulationyes  -1.2226     2.6427  -0.463
diabetesyes         -0.6479     2.4974  -0.259
dayd3               -8.8667     1.5028  -5.900
dayd5               -6.6667     1.5028  -4.436
dayd7               -3.3667     1.5028  -2.240
trtCAP              -0.6333     1.5028  -0.421
trthealthy          -0.5667     1.5028  -0.377
dayd3:trtCAP        -9.8000     2.1253  -4.611
dayd5:trtCAP        -7.0667     2.1253  -3.325
dayd7:trtCAP        -6.4667     2.1253  -3.043
dayd3:trthealthy     9.1333     2.1253   4.297
dayd5:trthealthy     7.9667     2.1253   3.748
dayd7:trthealthy     8.7667     2.1253   4.125
```

The point estimates from this output are quite similar to those from Figure 9: For the population-level effects, column `Estimate` of the output above needs to be compared to column `Estimate` from Figure 9. For the standard deviation of the group-level effects, the output above shows an estimate of ca. 3.17 whereas Figure 9 shows a posterior median of ca. 3.53. For the residual standard deviation, the output above shows an estimate of ca. 5.82 whereas Figure 9 shows a posterior median of ca. 5.84.

Frequentist 95 % CIs for all parameters may be produced by:

```
> ( prof_cis <- confint(lmm, quiet = TRUE, oldNames = FALSE) )
                            2.5 %     97.5 %
sd_(Intercept)|patID   1.4679897  4.1826203
sigma                  5.3286728  6.1802083
(Intercept)           21.0018188 57.9636424
age                   -0.1629532  0.3648855
anticoagulationyes    -5.6527729  3.2075731
diabetesyes           -4.8345008  3.5386306
dayd3                -11.7734425 -5.9598908
dayd5                 -9.5734425 -3.7598908
dayd7                 -6.2734425 -0.4598908
trtCAP                -3.5401092  2.2734425
trthealthy            -3.4734425  2.3401092
dayd3:trtCAP         -13.9108019 -5.6891981
dayd5:trtCAP         -11.1774685 -2.9558648
dayd7:trtCAP         -10.5774685 -2.3558648
dayd3:trthealthy       5.0225315 13.2441352
dayd5:trthealthy       3.8558648 12.0774685
dayd7:trthealthy       4.6558648 12.8774685
```

In general, these are quite similar to the Bayesian CrIs from Figure 9. However, there are some noticeable differences. For example, the standard deviation of the group-level effects for `patID` has a frequentist CI of ca. $(1.47, 4.18)$ and a Bayesian CrI of ca. $(2.00, 7.39)$. The large difference with respect to the upper boundary may be due to the fact that there are only $P = 10$ patients here, meaning that the marginal posterior of the `patID` standard deviation is only little informed by the data and more so by the weakly informative `student_t(3, 0, 7.4)` prior which **brms** (Bürkner, 2017, 2018) has chosen by default. This illustrates how the Bayesian prior guards against overfitting, at least to a certain extent.

Compared to the frequentist CIs, the Bayesian CrIs provide the big advantage that they have a more intuitive interpretation: They directly reflect our uncertainty concerning the parameters, after having seen the data and given our prior. In contrast, the frequentist CIs have a quite complicated interpretation. For example, take the `patID` standard deviation already mentioned above. Its 95 % CI of ca. $(1.47, 4.18)$ means that among an infinite number of hypothetical replications of this dermatological study, the interval constructed this way would cover the true `patID` standard deviation at least 95 % of the time. In case of highly informative data (and a prior which is not too informative), one may argue that the frequentist results should be similar to those from a Bayesian analysis so that the frequentist results may be interpreted (approximately) in a Bayesian way, but this argument does not hold here where we have only $P = 10$ patients.

The frequentist CIs presented above are profile CIs. Apart from profile CIs, **lme4** (more precisely, `lme4::confint.merMod()`) also offers Wald and bootstrap CIs. For the Wald CIs, the sampling distributions of the coefficient estimators are approximated by normal distributions. This large-sample approximation is problematic at least for the patient-specific predictors (`age`, `anticoagulationyes`, and `diabetesyes`) since we have only $P = 10$ patients here. Note that even though the profile CIs relax the assumptions required for the Wald CIs, they also make use of a large-sample approximation (Bates et al., 2015). As mentioned in section "Introduction" of the main article, the Bayesian CrIs from Figure 9 do not require such a large-sample approximation. Although the bootstrap CIs also do not require a large-sample approximation, their frequentist performance usually suffers for small sample sizes, too (Hesterberg, 2015). This is also the case (even if not that pronounced) for the parametric bootstrap that **lme4** uses (Scholz, 2007). The existence of three different CI methods (profile, Wald, and bootstrap CIs) demonstrates another advantage of Bayesian statistics mentioned in section "Introduction", namely the fact that MCMC sampling is a generic inference method suitable for most practical cases.

The output from tab "Custom summary" (Figure 10) can be achieved in a frequentist context by the help of the **multcomp** package (Hothorn et al., 2008):

```
> coefs <- fixef(lmm)
> ncoefs <- length(coefs)
> contr_mat <- matrix(0, nrow = length(unique(CAP$day)), ncol = ncoefs,
+                     dimnames = list(paste0("CAP_", unique(CAP$day)),
+                                     names(coefs)))
> contr_mat[, "trtCAP"] <- 1
> for (day_i in setdiff(unique(CAP$day), "d1")) {
+   contr_mat[paste0("CAP_", day_i), paste0("day", day_i, ":trtCAP")] <- 1
+ }
> library(multcomp)
> linhyp <- glht(lmm, linfct = contr_mat)
> confint(linhyp, calpha = univariate_calpha())

         Simultaneous Confidence Intervals

Fit: lmer(formula = TWI ~ age + anticoagulation + diabetes + day *
    trt + (1 | patID), data = CAP)

Quantile = 1.96
95% confidence level


Linear Hypotheses:
            Estimate lwr      upr
CAP_d1 == 0  -0.6333  -3.5788   2.3121
CAP_d3 == 0 -10.4333 -13.3788  -7.4879
CAP_d5 == 0  -7.7000 -10.6455  -4.7545
```

```
CAP_d7 == 0  -7.1000 -10.0455  -4.1545
```

These point and interval estimates are quite close to their Bayesian counterparts from Figure 10, but in terms of the interval estimates, the Bayesian CrIs again have a more intuitive interpretation than the frequentist CIs presented here. Also note that the **multcomp** CIs above are Wald CIs, i.e., they rely on a normal approximation.

The "Custom summary" also shows another advantage of the Bayesian analysis, namely the easy propagation of posterior uncertainty into derived quantities. Of course, this is less important for the Gaussian family with the identity link function used here, but it becomes important for families with non-identity link functions. In other words, even if frequentist methods might be able to deal with linear functions of the parameters, they usually struggle with nonlinear functions. In a Bayesian analysis, one simply has to apply the transformation—be it linear or not—to the parameter draws to obtain transformed parameter draws which incorporate the uncertainty inherent to the original draws.

The conditional-effects plot from Figure 11 can be reproduced in a frequentist context by the help of packages **emmeans** (Lenth, 2022) and **ggplot2** (Wickham, 2016), for example:

```
> library(emmeans)
> rg <- ref_grid(lmm, at = list(anticoagulation = "no", diabetes = "no"))
> em <- emmeans(rg, specs = c("day", "trt"))
> plem <- plot(em, plotit = FALSE)
> library(ggplot2)
> dg <- position_dodge(width = 0.2)
> ggplot(plem, aes(x = day, y = the.emmean, color = trt, group = trt)) +
+   geom_point(position = dg) +
+   ylab("TWI") +
+   geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),
+                 position = dg,
+                 width = 0.25)
```

which results in Figure S1. Note that the CIs in this plot make use of the Kenward-Roger method (Kenward and Roger, 1997). Especially for small sample sizes, Kenward-Roger CIs are preferable to normal-approximation CIs, but in general, they are still based on an approximation to the true sampling distribution (Kenward and Roger, 1997). So the argument of avoiding approximations through a Bayesian analysis still holds when disregarding minor approximations induced by MCMC sampling.

Like the "Custom summary" tab, the "Conditional effects" tab also demonstrates the easy propagation of posterior uncertainty into derived quantities, even though in this example, we again have only linear transformations of the parameters, so the strength of the Bayesian approach does not stand out.

The closest we could get to the PPCs (Figures 12 and 13) from a frequentist perspective was by the help of package **performance** (Lüdecke et al., 2021). A frequentist version of the PPC *via* overlaid kernel density estimates may be achieved by:

```
> library(performance)
> theme_set(theme_bw())
> set.seed(847299)
> check_predictions(lmm, iterations = 8)
```

which results in Figure S2. A frequentist version of the PPC *via* summary statistics may be achieved by:

```
> check_predictions(lmm, iterations = 4000, check_range = TRUE)
```

which results in Figure S3. However, the crucial difference to the Bayesian PPCs is that in these frequentist PPC versions, the uncertainty with respect to parameter estimation is not

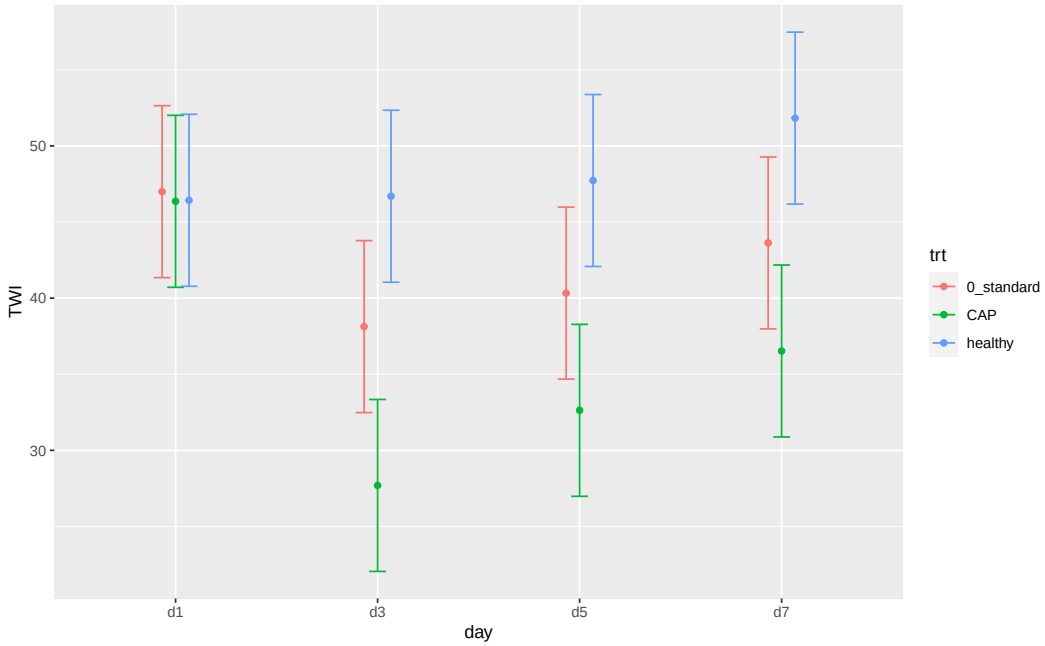

Figure S1: Frequentist conditional-effects plot. The CIs in this plot are Kenward-Roger CIs (Kenward and Roger, 1997). This frequentist plot largely agrees with the Bayesian one from Figure 11, but differs in terms of interpretation and generalizability (see text).

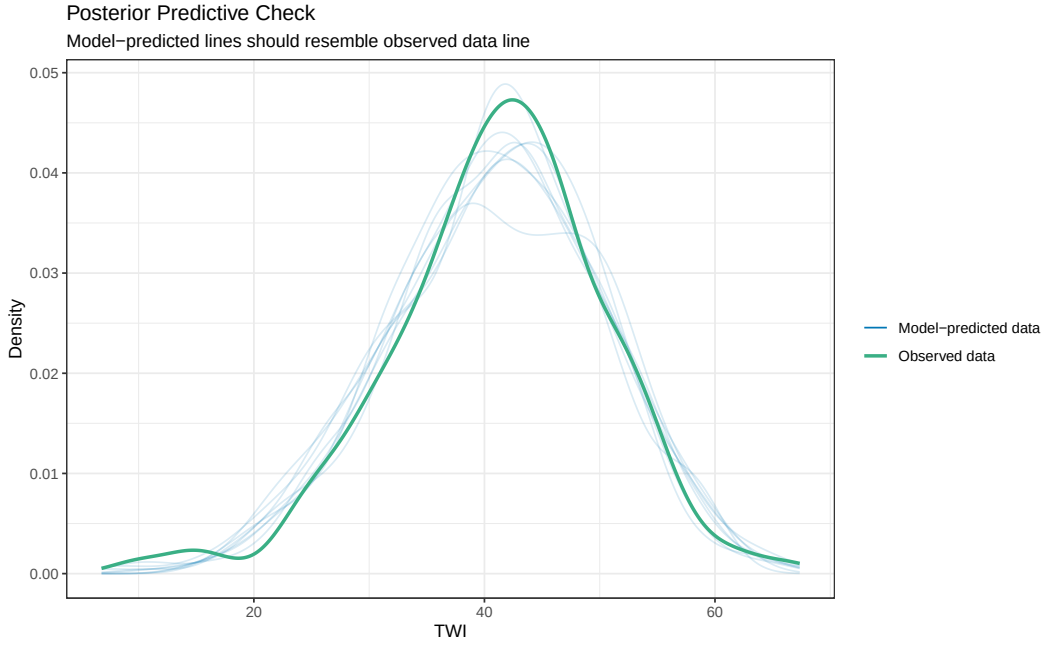

Figure S2: Frequentist version of the PPC *via* overlaid kernel density estimates. The green density line corresponds to the observed outcome values whereas each of the 8 overlaid blue density lines corresponds to one dataset simulated under the model's parameter estimates. The outer appearance of this frequentist plot is roughly the same as that of the Bayesian one from Figure 12, but a major difference is that this frequentist plot does not reflect the uncertainty from parameter estimation.

taken into account, which is a central feature of the Bayesian PPC. This demonstrates again the easy propagation of posterior uncertainty into derived quantities.

Finally, analogously to the Bayesian CrIs from Figure 14, the frequentist CIs calculated above can be visualized by the following code which uses **ggplot2** again (note that we only plot the profile CIs and only for those parameters which are also shown in Figure 14):

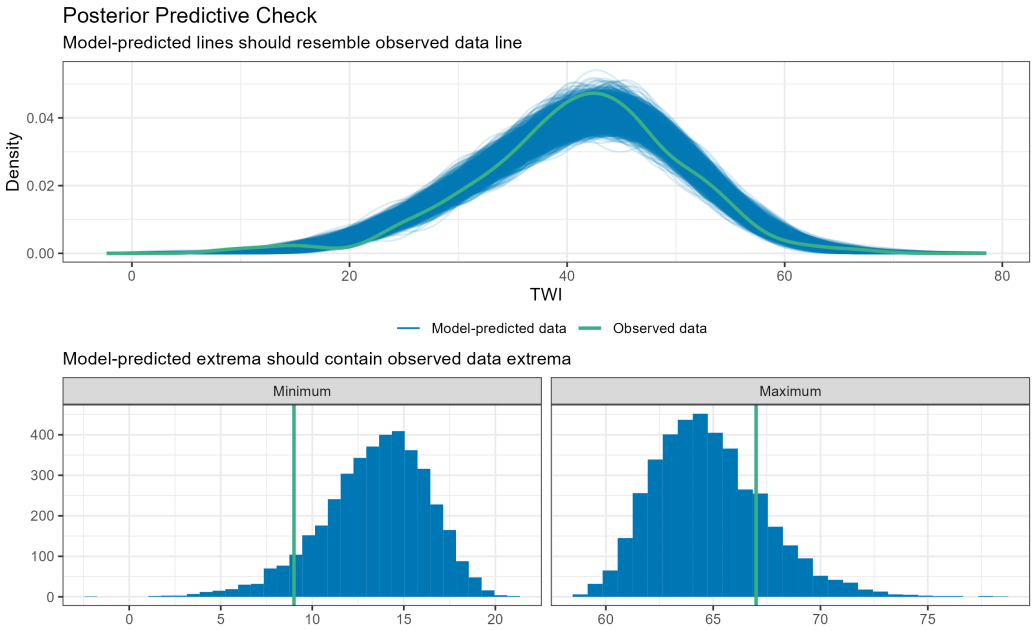

Figure S3: Frequentist version of the PPC *via* summary statistics. In contrast to Figure 13, mean and standard deviation are missing as summary statistics. For minimum and maximum, the same conclusions hold as for Figure 13. The upper plot (basically the same as Figure S2, but with more iterations) was added automatically by `performance::check_predictions()` and could not be removed easily.

```
> point_estims <- c(sd_patID = attr(VarCorr(lmm)$patID, "stddev"),
+                   sigma = sigma(lmm),
+                   fixef(lmm))
> prof_cis_gg <- cbind(as.data.frame(prof_cis),
+                      point_est = point_estims,
+                      Parameter = rownames(prof_cis))
> pars_sel <- grep("^sigma$|^sd_|:trthealthy",
+                  row.names(prof_cis_gg),
+                  invert = TRUE)
> prof_cis_gg <- prof_cis_gg[pars_sel, ]
> prof_cis_gg <- within(prof_cis_gg, {
+   Parameter <- factor(Parameter, levels = rev(unique(Parameter)))
+ })
> ggplot(prof_cis_gg, aes(y = Parameter)) +
+   geom_point(aes(x = point_est)) +
+   xlab("") +
+   geom_errorbar(aes(xmin = `2.5 %`, xmax = `97.5 %`),
+                 width = 0.25)
```

which gives Figure S4. Comparing this with Figure 14 shows again the rough similarity of the interval estimates, but with a more convenient interpretation of the Bayesian CrIs.

Note that unlike the frequentist CI plot, the Bayesian CrI plot also features *inner* intervals. The reason is that the marginal posteriors do not need to follow a closed-form distribution, so it makes sense to include as much information about them as possible. The kernel density estimates for the marginal posteriors available in the **shinystan** (Gabry, 2022) app (not shown here) also remind us of this flexibility.

Finally, we point out that in the Bayesian analysis from section "Example", we did not really make use of the advantage that a Bayesian analysis can incorporate prior information, e.g., from expert knowledge (we picked the `student_t(3, 0, 30)` prior for the regression coefficients only for demonstrative purposes and left all other priors at their defaults). In a

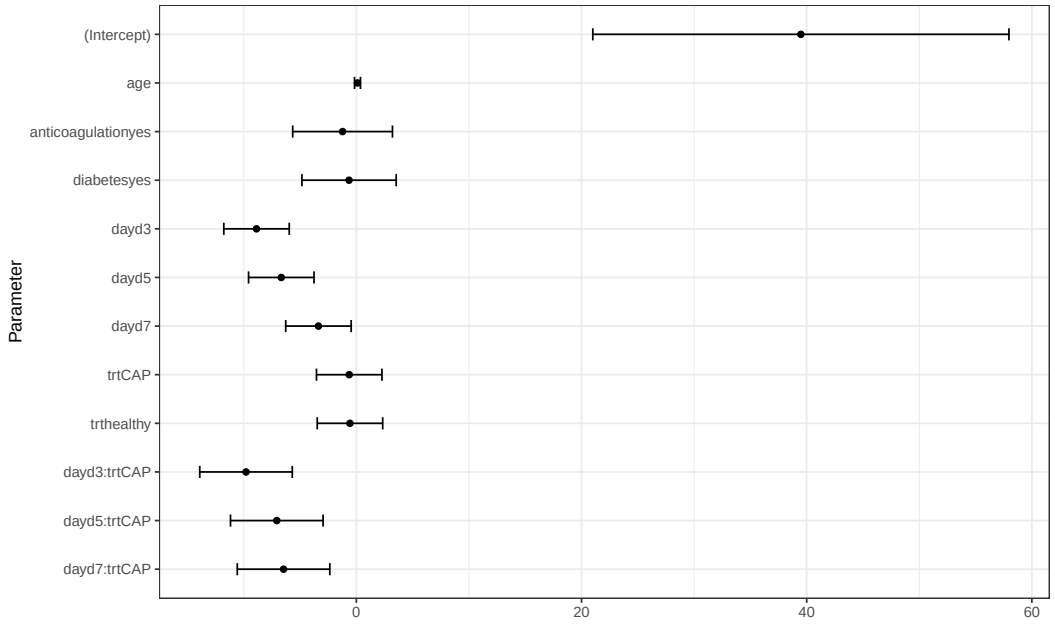

Figure S4: Frequentist 95 % profile CIs. This plot is restricted to those parameters which are also shown in Figure 14. As may be seen from this plot, the frequentist CIs are quite similar to the Bayesian CrIs (Figures 9 and 14). However, their interpretation differs (see text).

real-world application where such information is available, it should be incorporated.

Bibliography

J. L. Arbuckle. *Amos*. IBM SPSS, Chicago, IL, USA, 2020. Version 27.0. [p2]

D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. URL https://doi.org/10.18637/jss.v067.i01. [p4, 6]

C. Belitz, A. Brezger, N. Klein, T. Kneib, S. Lang, and N. Umlauf. *BayesX - Software for Bayesian Inference in Structured Additive Regression Models*, 2015. URL http://www.bayesx.org. Version 3.0.2. [p1]

BEsmarter Team. *BEsmarter*, 2020a. URL https://besmarter-team.shinyapps.io/BEsmarter-GUI/. Web application (shiny app). Accessed on October 15, 2020. [p3]

BEsmarter Team. BEsmarter source code, 2020b. URL https://github.com/besmarter/BSTApp. Accessed on April 21, 2020. [p3]

A. Brezger, T. Kneib, and S. Lang. BayesX: Analyzing Bayesian structural additive regression models. *Journal of Statistical Software*, 14(11):1–22, 2005. URL https://doi.org/10.18637/jss.v014.i11. [p1]

P.-C. Bürkner. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28, 2017. URL https://doi.org/10.18637/jss.v080.i01. [p5]

P.-C. Bürkner. Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1):395–411, 2018. URL https://doi.org/10.32614/RJ-2018-017. [p5]

W. Chang, J. Cheng, J. J. Allaire, C. Sievert, B. Schloerke, Y. Xie, J. Allen, J. McPherson, A. Dipert, and B. Borges. *shiny: Web Application Framework for R*, 2022. URL https://CRAN.R-project.org/package=shiny. R package, version 1.7.2. [p1]

P. D. Congdon. *Bayesian Hierarchical Models: With Applications Using R*. CRC Press, Boca Raton, FL, USA, 2nd edition, 2020. ISBN 978-1-4987-8575-4. URL https://doi.org/10.1201/9780429113352. [p1]

G. Emvalomatis. *BayES: Bayesian Econometrics Software*, 2020. URL https://bayeconsoft.com/. Version 2.5. Accessed on November 24, 2020. [p3]

J. Gabry. *shinystan: Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models*, 2022. URL https://mc-stan.org/shinystan/. R package, version 2.6.0. [p9]

W. R. Gilks, A. Thomas, and D. J. Spiegelhalter. A language and program for complex Bayesian modelling. *Journal of the Royal Statistical Society, Series D (The Statistician)*, 43(1): 169–177, 1994. URL https://doi.org/10.2307/2348941. [p1]

R. L. Grant, B. Carpenter, D. C. Furr, and A. Gelman. Introducing the StataStan interface for fast, complex Bayesian modeling using Stan. *The Stata Journal*, 17(2):330–342, 2017a. URL https://doi.org/10.1177/1536867X1701700205. [p3]

R. L. Grant, D. C. Furr, B. Carpenter, and A. Gelman. Fitting Bayesian item response models in Stata and Stan. *The Stata Journal*, 17(2):343–357, 2017b. URL https://doi.org/10.1177/1536867X1701700206. [p3]

T. C. Hesterberg. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. *The American Statistician*, 69(4):371–386, 2015. URL https://doi.org/10.1080/00031305.2015.1089789. [p6]

T. Hothorn, F. Bretz, and P. Westfall. Simultaneous inference in general parametric models. *Biometrical Journal*, 50(3):346–363, 2008. URL https://doi.org/10.1002/bimj.200810425. [p6]

IBM Corp. *IBM SPSS Statistics for Windows*. IBM Corp., Armonk, NY, USA, 2020. Version 27.0. [p3]

JASP Team. Announcing JASP 0.12, 2020a. URL https://jasp-stats.org/2020/04/10/announcing-jasp-0-12/. Accessed on April 21, 2020. [p2]

JASP Team. The JASP statistics project, 2020b. URL https://github.com/jasp-stats. Accessed on April 21, 2020. [p2]

JASP Team. R package list, 2020c. URL https://jasp-stats.org/r-package-list/. Accessed on April 21, 2020. [p2]

JASP Team. *JASP*, 2022. URL https://jasp-stats.org/. Version 0.16.3. [p2]

G. Karabatsos. A menu-driven software package for Bayesian regression analysis. *The ISBA Bulletin*, 22(4):13–16, 2015. [p2]

G. Karabatsos. A menu-driven software package of Bayesian nonparametric (and parametric) mixed models for regression analysis and density estimation. *Behavior Research Methods*, 49(1):335–362, 2017. URL https://doi.org/10.3758/s13428-016-0711-7. [p2]

M. G. Kenward and J. H. Roger. Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 53(3):983–997, 1997. URL https://doi.org/10.2307/2533558. [p7, 8]

R. V. Lenth. *emmeans: Estimated Marginal Means, aka Least-Squares Means*, 2022. URL https://CRAN.R-project.org/package=emmeans. R package, version 1.7.5. [p7]

D. Lunn, D. Spiegelhalter, A. Thomas, and N. Best. The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25):3049–3067, 2009. URL https://doi.org/10.1002/sim.3680. [p1]

D. Lunn, C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter. *The BUGS Book: A Practical Introduction to Bayesian Analysis*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, Boca Raton, FL, USA, 2012. ISBN 978-1-58488-849-9. URL https://doi.org/10.1201/b13613. [p1]

D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337, 2000. URL https://doi.org/10.1023/A:1008929526011. [p1]

D. Lüdecke, M. S. Ben-Shachar, I. Patil, P. Waggoner, and D. Makowski. performance: An R package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, 6(60):3139, 2021. URL https://doi.org/10.21105/joss.03139. [p7]

B. B. McShane, D. Gal, A. Gelman, C. Robert, and J. L. Tackett. Abandon statistical significance. *The American Statistician*, 73:235–245, 2019. URL https://doi.org/10.1080/00031305.2018.1527253. [p3]

Microsoft Corporation. *Microsoft Excel*, 2016. URL https://office.microsoft.com/excel. Version 2016 (16.0). [p1]

M. Plummer. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In K. Hornik, F. Leisch, and A. Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, Vienna, Austria, 2003. URL https://www.r-project.org/conferences/DSC-2003/Proceedings/. [p1]

M. Plummer. *JAGS Version 4.3.0 User Manual*, 2017. URL https://sourceforge.net/projects/mcmc-jags/files/Manuals/4.x/jags_user_manual.pdf. Accessed on May 8, 2020. [p1]

H. Qian. *Toolkit on Econometrics and Economics Teaching*. MATLAB Central File Exchange, 2011. URL https://www.mathworks.com/matlabcentral/fileexchange/32601-toolkit-on-econometrics-and-economics-teaching. MATLAB package, version from August 19, 2011. Accessed on March 26, 2020. [p2]

A. Ramírez-Hassan and M. Graciano-Londoño. A GUIded tour of Bayesian regression. *The R Journal*, 13(2):135–152, 2021. URL https://doi.org/10.32614/RJ-2021-081. [p3]

D. B. Rubin. The Bayesian bootstrap. *The Annals of Statistics*, 9(1):130–134, 1981. URL https://doi.org/10.1214/aos/1176345338. [p3]

F. W. Scholz. The bootstrap small sample properties. Technical report, University of Washington, 2007. URL https://faculty.washington.edu/fscholz/Reports/bootstrap-report.pdf. Edited version of a technical report of same title and issued as "bcstech-93-051", Boeing Computer Services, Research and Technology. [p6]

D. Spiegelhalter, A. Thomas, N. Best, and D. Lunn. *OpenBUGS User Manual, Version 3.2.3*, 2014. URL http://www.openbugs.net/Manuals/Manual.html. Accessed on March 25, 2020. [p1]

StataCorp. *Stata*. StataCorp LLC, College Station, TX, USA, 2019a. Release 16. [p2]

StataCorp. *Stata Bayesian Analysis Reference Manual*. Stata Press, College Station, TX, USA, 2019b. ISBN 978-1-59718-272-0. URL https://www.stata.com/manuals/bayes.pdf. Release 16. Accessed on May 7, 2020. [p2, 3]

The MathWorks, Inc. *MATLAB*. The MathWorks, Inc., Natick, MA, USA, 2020a. Release 2020a. [p2]

The MathWorks, Inc. *MATLAB Compiler*. The MathWorks, Inc., Natick, MA, USA, 2020b. Release 2020a. [p2]

C. Wang, T. A. Louis, N. C. Henderson, C. O. Weiss, and R. Varadhan. beanz: An R package for Bayesian analysis of heterogeneous treatment effects with a graphical user interface. *Journal of Statistical Software*, 85(7):1–31, 2018. URL https://doi.org/10.18637/jss.v085.i07. [p1]

F. Weber. *shinybrms: Graphical User Interface ('shiny' App) for 'brms'*, 2022. URL https://fweber144.github.io/shinybrms/. R package, version 1.8.0. [p2]

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Use R. Springer, New York, NY, USA, 2nd edition, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org. [p7]

P. Woodward. *Bayesian Analysis Made Simple: An Excel GUI for WinBUGS*. Chapman & Hall/CRC Biostatistics Series. CRC Press, Boca Raton, FL, USA, 2011. ISBN 978-1-4398-3954-6. URL https://doi.org/10.1201/b11235. [p1]

Y. Xie and J. Wang. *iBUGS: An Interface to WinBUGS/OpenBUGS/JAGS by gWidgets*, 2013. URL https://CRAN.R-project.org/package=iBUGS. R package, version 0.1.4. [p1]