

# An introduction to Bayesian multilevel modeling with brms

---

Paul Bürkner

# The Bayes Theorem

$$p(\theta | y) = \frac{p(y | \theta) p(\theta)}{p(y)}$$

$$p(\theta | y) \propto p(y | \theta) p(\theta) = p(y, \theta)$$

# Advantages and Disadvantages of Bayesian Statistics

## Advantages:

- Natural approach to expressing uncertainty
- Ability to incorporate prior information
- Increased modeling flexibility
- Full posterior distribution of parameters
- Natural propagation of uncertainty

## Disadvantages:

- Slow Speed of model estimation



## Stan syntax: Linear Regression

```
data {  
  int<lower=1> N;  // total number of observations  
  vector[N] Y;    // response variable  
  int<lower=1> K;  // number of regression coefficients  
  matrix[N, K] X; // predictor design matrix  
}  
  
parameters {  
  vector[K] b;    // regression coefficients  
  real<lower=0> sigma; // residual SD  
}  
  
model {  
  vector[N] mu;  
  mu = X * b;  
  sigma ~ exponential(0.1);  
  Y ~ normal(mu, sigma);  
}
```

# Bayesian Software: brms



- Specify models via extended R formula syntax
- Internally write Stan code that is readable yet fast
- Provide an easy interface for defining priors
- Facilitate post-processing

# Stan syntax: Simple multilevel model by brms (1)

```
functions {  
}  
data {  
  int<lower=1> N; // total number of observations  
  vector[N] Y; // response variable  
  int<lower=1> K; // number of population-level effects  
  matrix[N, K] X; // population-level design matrix  
  // data for group-level effects of ID 1  
  int<lower=1> J_1[N];  
  int<lower=1> N_1;  
  int<lower=1> M_1;  
  vector[N] Z_1_1;  
  vector[N] Z_1_2;  
  int<lower=1> NC_1;  
  int prior_only; // should the likelihood be ignored?  
}  
transformed data {  
  int Kc;  
  matrix[N, K - 1] Xc; // centered version of X  
  vector[K - 1] means_X; // column means of X before centering  
  Kc = K - 1; // the intercept is removed from the design matrix  
  for (i in 2:K) {  
    means_X[i - 1] = mean(X[, i]);  
    Xc[, i - 1] = X[, i] - means_X[i - 1];  
  }  
}
```



## Stan syntax: Simple multilevel model by brms (2)

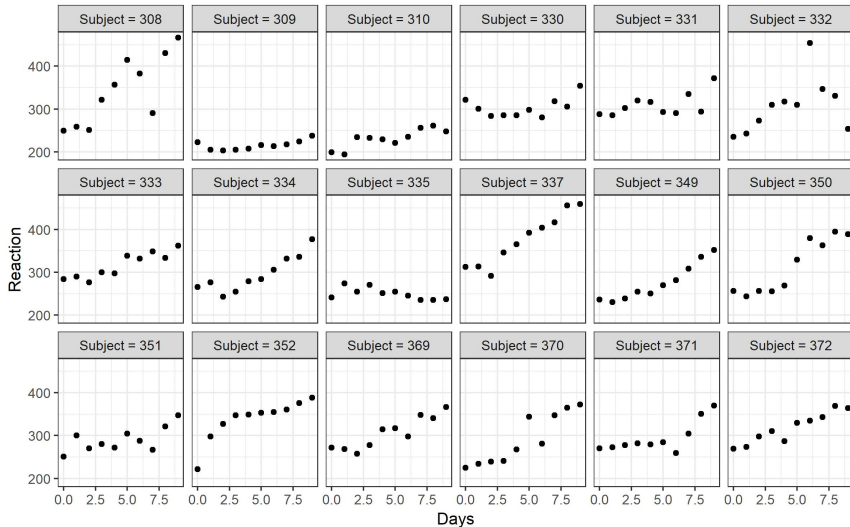
```
parameters {  
  vector[Kc] b; // population-level effects  
  real temp_Intercept; // temporary intercept  
  real<lower=0> sigma; // residual SD  
  vector<lower=0>[M_1] sd_1; // group-level standard deviations  
  matrix[M_1, N_1] z_1; // unscaled group-level effects  
  // cholesky factor of correlation matrix  
  cholesky_factor_corr[M_1] L_1;  
}  
transformed parameters {  
  // group-level effects  
  matrix[N_1, M_1] r_1;  
  vector[N_1] r_1_1;  
  vector[N_1] r_1_2;  
  r_1 = (diag_pre_multiply(sd_1, L_1) * z_1)';  
  r_1_1 = r_1[, 1];  
  r_1_2 = r_1[, 2];  
}
```

## Stan syntax: Simple multilevel model by brms (3)

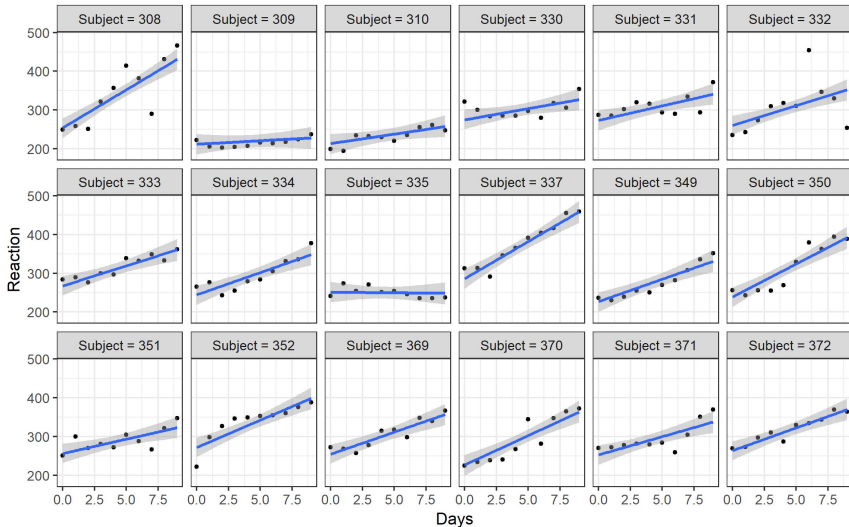
```
model {  
  vector[N] mu;  
  mu = Xc * b + temp_Intercept;  
  for (n in 1:N) {  
    mu[n] = mu[n] + (r_1_1[J_1[n]]) * Z_1_1[n] + (r_1_2[J_1[n]]) * Z_1_2[n];  
  }  
  // prior specifications  
  sigma ~ student_t(3, 0, 56);  
  sd_1 ~ student_t(3, 0, 56);  
  L_1 ~ lkj_corr_cholesky(1);  
  to_vector(z_1) ~ normal(0, 1);  
  // likelihood contribution  
  if (!prior_only) {  
    Y ~ normal(mu, sigma);  
  }  
}  
  
generated quantities {  
  real b_Intercept; // population-level intercept  
  corr_matrix[M_1] Cor_1;  
  vector<lower=-1,upper=1>[NC_1] cor_1;  
  b_Intercept = temp_Intercept - dot_product(means_X, b);  
  // take only relevant parts of correlation matrix  
  Cor_1 = multiply_lower_tri_self_transpose(L_1);  
  cor_1[1] = Cor_1[1,2];  
}
```

We should think about data structure

# Example: Effects of Sleep Deprivation on Reaction Times

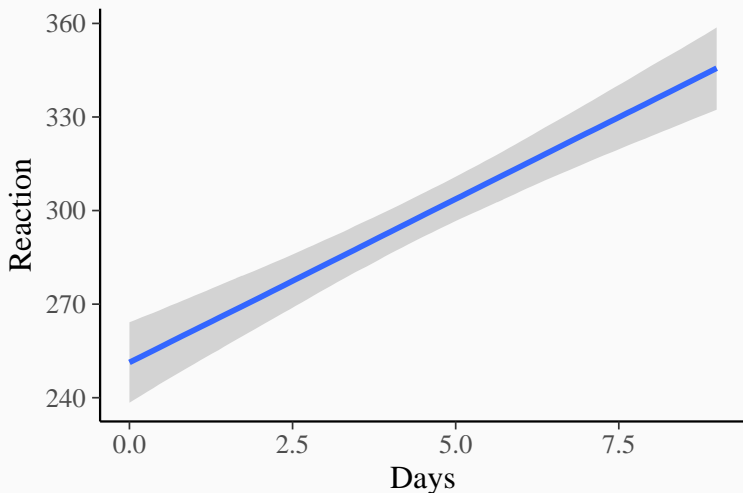


# Regression Lines for Specific Subjects



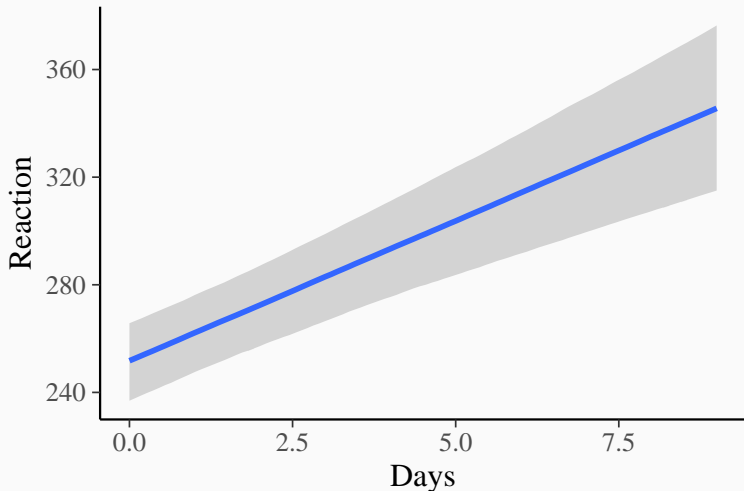
## Linear Regression with brms

```
fit_sleep1 <- brm(Reaction ~ Days, data = sleepstudy)
conditional_effects(fit_sleep1)
```



## Multilevel Models with brms

```
form2 <- Reaction ~ 1 + Days + (1 + Days | Subject)
fit_sleep2 <- brm(form2, data = sleepstudy)
```

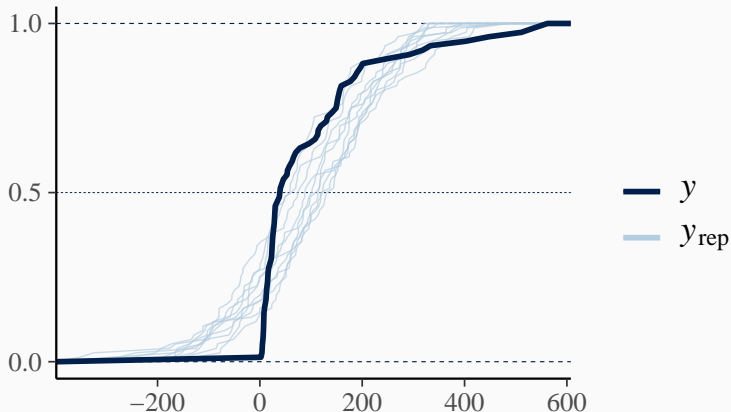


We should think about distributions



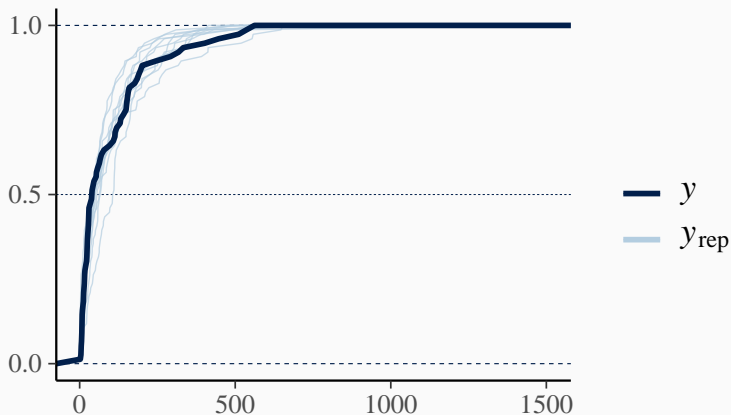
## We should think about the likelihood

```
fit_kidney1 <- brm(time ~ age + sex, family = gaussian())  
pp_check(fit_kidney1, type = "ecdf_overlay")
```



## We should think about the likelihood

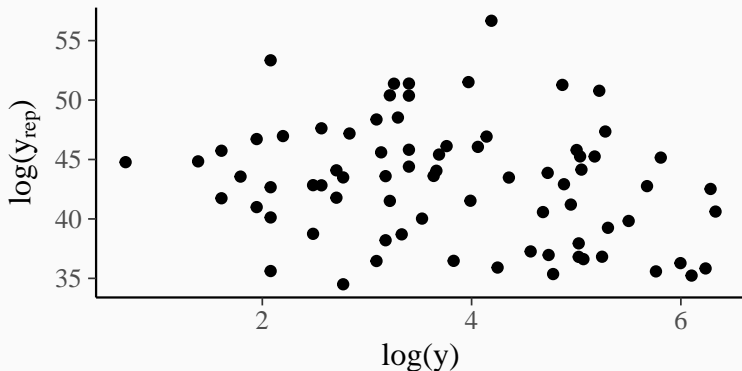
```
fit_kidney2 <- brm(time ~ age + sex, family = Gamma("log"))  
pp_check(fit_kidney2, type = "ecdf_overlay")
```



## We should think about the prior

```
fit_kidney3 <- brm(  
  time ~ age + sex, family = Gamma("log"),  
  prior = prior(normal(0, 0.5)),  
  sample_prior = "only")
```

Prior predictions:



# Censoring in brms

```
brm(time | cens(censored) ~ age + sex, ...)
```

- `cens()` is called an addition term in brms
- `censored` is the variable in the data that indicates censoring
  - 0: if the observation is not censored
  - 1: if the observation is right censored
  - -1: if the observation is left censored
  - 2: if the observation is interval censored

## Modeling of unknown non-linear functions

$$y = f(x) + \varepsilon$$

# Splines and Gaussian Processes

Splines:

$$f(x) = \sum_{j=1}^J \beta_j b_j(x)$$

$$\beta_j \sim D(\lambda)$$

```
brm(y ~ s(x) + ...)
```

Gaussian Processes:

$$f(x) \sim \text{Normal}(0, K(x, \alpha))$$

```
brm(y ~ gp(x) + ...)
```

# Housing Rents in Munich

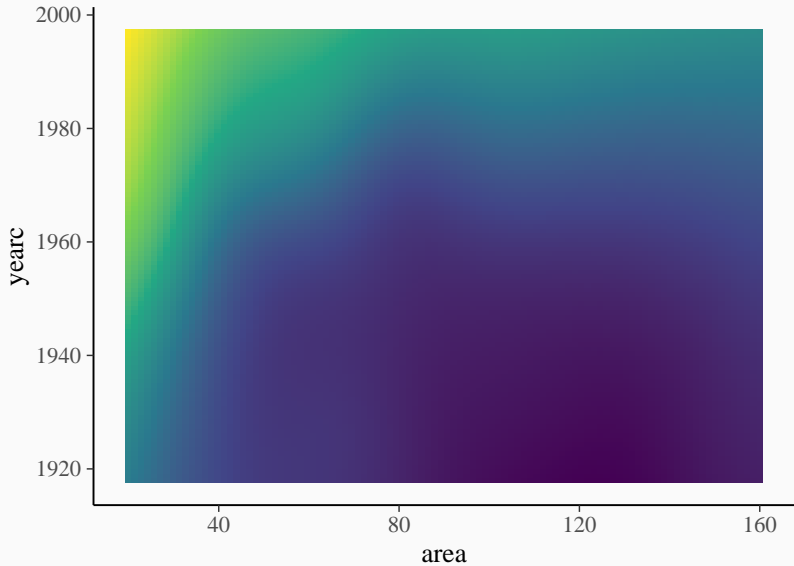
Predicting  $\mu$ :

```
bform1 <- bf(rentsqm ~ s(area, yearc) + (1 | district))  
fit_rent1 <- brm(bform1, ...)
```

Predicting  $\mu$  and  $\sigma$ :

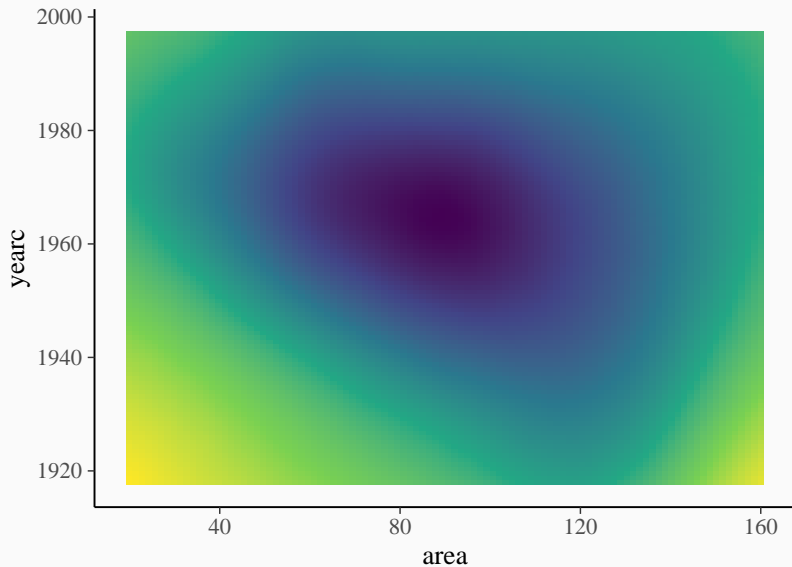
```
bform2 <- bf(  
  rentsqm ~ s(area, yearc) + (1 | d | district),  
  sigma ~ s(area, yearc) + (1 | d | district)  
)  
fit_rent2 <- brm(bform2, ...)  
  
conditional_smooths(fit_rent2, stype = "raster")
```

## Housing Rents in Munich: Predictions of $\mu$





# Housing Rents in Munich: Predictions of $\sigma$



# Bayesian Cross Validation

How do we estimate predictions for new data without new data?

Cross Validation (CV):

$$p(y_S | y_{-S}) = \int p(y_S | \theta) p(\theta | y_{-S}) d\theta$$

Expected Log Predictive Density (ELPD):

$$\text{ELPD} = \sum_{S \in \Sigma} \log p(y_S | y_{-S})$$

Evaluates **Out-of-Sample Fit** and penalizes **Posterior Complexity**

# Approximate Leave-One-Out Cross-Validation

How can we make cross-validation feasible for Bayesian models?

Approximate Leave-One-Out Cross-Validation (LOO-CV):

$$p(y_i | y_{-i}) \approx \int p(y_i | \theta) \tilde{p}(\theta | y) d\theta$$

```
loo(fit_rent2)

##
## Computed from 2000 by 3082 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo  -6455.2 41.7
## p_loo      202.2  7.1
## looic      12910.4 83.3
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##           Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)   3038 98.6%   227
## (0.5, 0.7]  (ok)      41  1.3%   156
## (0.7, 1]    (bad)      3  0.1%    72
## (1, Inf)    (very bad) 0  0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

## Housing rents: Does modeling $\sigma$ improve predictions?

Compare the model with and without prediction of  $\sigma$ :

```
loo_compare(loo(fit_rent1), loo(fit_rent2))
```

##		elpd_diff	se_diff
##	fit_rent2	0.0	0.0
##	fit_rent1	-50.8	10.6

## Case Study: Treatment of Epilepsy

```
data("epilepsy", package = "brms")
```

count	Age	Base	Trt	patient	visit
5	31	11	0	1	1
3	30	11	0	2	1
2	25	6	0	3	1
4	36	8	0	4	1
7	22	66	0	5	1
5	29	27	0	6	1
6	31	12	0	7	1
40	42	52	0	8	1
5	37	23	0	9	1
14	28	10	0	10	1

## Epilepsy: Bayesian Model Building (1)

```
fit_epi1 <- brm(  
  count ~ Age + Base * Trt,  
  data = epilepsy,  
  file = "models/fit_epi1"  
)
```

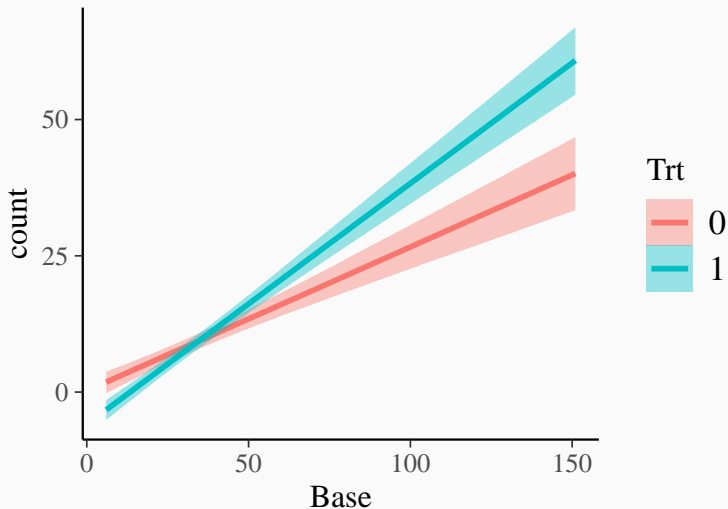
# Numerical Summary

```
summary(fit_epil)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: count ~ Age + Base * Trt
## Data: epilepsy (Number of observations: 236)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -7.11      2.63  -12.25   -1.98 1.00    3799    2671
## Age           0.26      0.08    0.09    0.42 1.00    3638    2330
## Base          0.26      0.03    0.21    0.32 1.00    2275    2908
## Trt1         -6.17      1.53   -9.08   -3.12 1.00    2118    2702
## Base:Trt1     0.18      0.04    0.10    0.25 1.00    1944    2668
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma        7.63      0.35    6.99    8.36 1.00    3842    2729
##
## Samples were drawn using sampling(NUTS). 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).
```

## Graphical Summary

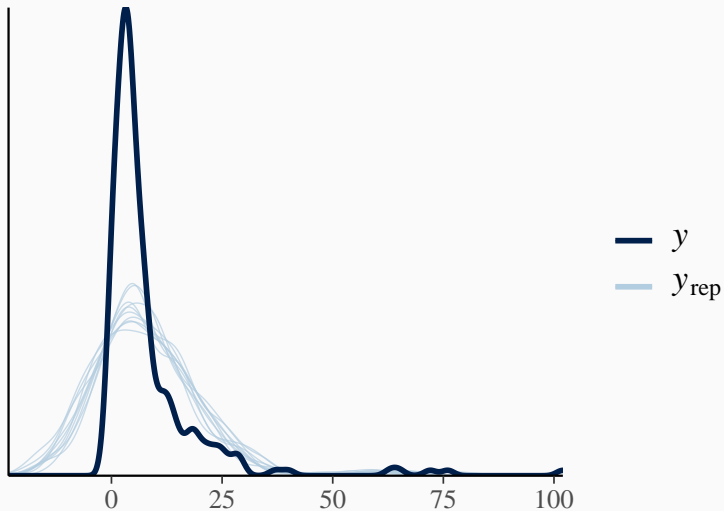
```
conditional_effects(fit_epi1, "Base:Trt")
```





# Posterior-Predictive Checks

```
pp_check(fit_epi1)
```



## Epilepsy: Bayesian Model Building (2)

```
fit_epi2 <- brm(  
  count ~ Age + Base * Trt,  
  data = epilepsy,  
  family = poisson("log"),  
  file = "models/fit_epi2"  
)  
  
fit_epi2 <- add_criterion(fit_epi2, "loo")
```

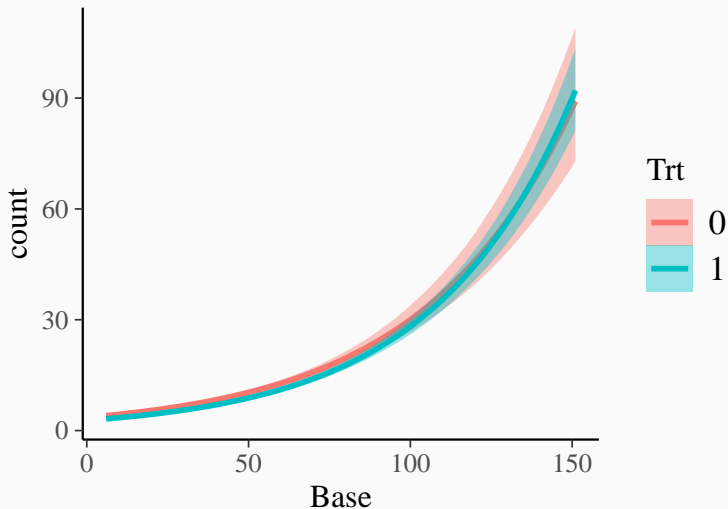
# Numerical Summary

```
summary(fit_epi2)
```

```
## Family: poisson
## Links: mu = log
## Formula: count ~ Age + Base * Trt
## Data: epilepsy (Number of observations: 236)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.59      0.14   0.32   0.87 1.00    2212    2017
## Age             0.02      0.00   0.02   0.03 1.00    2561    2150
## Base            0.02      0.00   0.02   0.02 1.00    2660    2670
## Trt1           -0.25      0.08  -0.40  -0.10 1.00    2027    1817
## Base:Trt1       0.00      0.00  -0.00   0.00 1.00    2485    2746
##
## Samples were drawn using sampling(NUTS). 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).
```

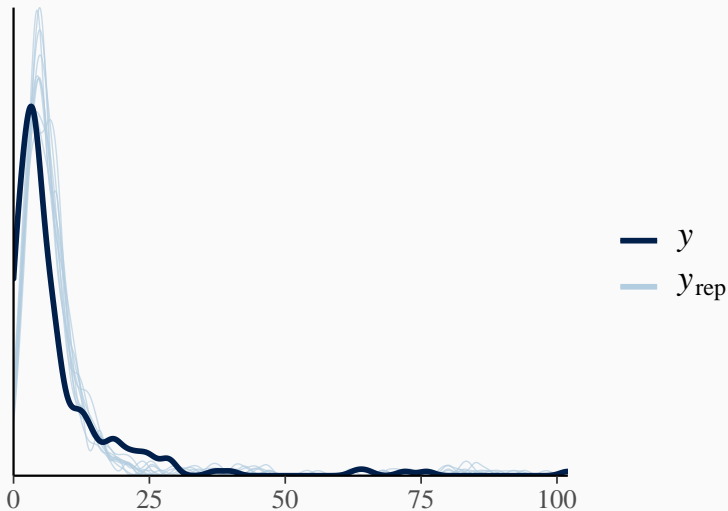
## Graphical Summary

```
conditional_effects(fit_epi2, "Base:Trt")
```



# Posterior-Predictive Checks

```
pp_check(fit_epi2)
```



# Leave-One-Out Cross-Validation

```
loo(fit_epi2)

##
## Computed from 4000 by 236 log-likelihood matrix
##
##      Estimate      SE
## elpd_loo   -874.8  90.1
## p_loo       22.8   5.5
## looic      1749.5 180.2
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##
##      Count Pct.    Min. n_eff
## (-Inf, 0.5] (good)   233  98.7%   875
## (0.5, 0.7]  (ok)     2    0.8%   154
## (0.7, 1]    (bad)     1    0.4%    86
## (1, Inf)    (very bad) 0    0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

## Epilepsy: Bayesian Model Building (3)

```
fit_epi3 <- brm(  
  count ~ zAge + zBase * Trt + (1 | patient),  
  data = epilepsy,  
  family = poisson("log"),  
  file = "models/fit_epi3"  
)  
  
fit_epi3 <- add_criterion(fit_epi3, "loo")
```

# Numerical Summary

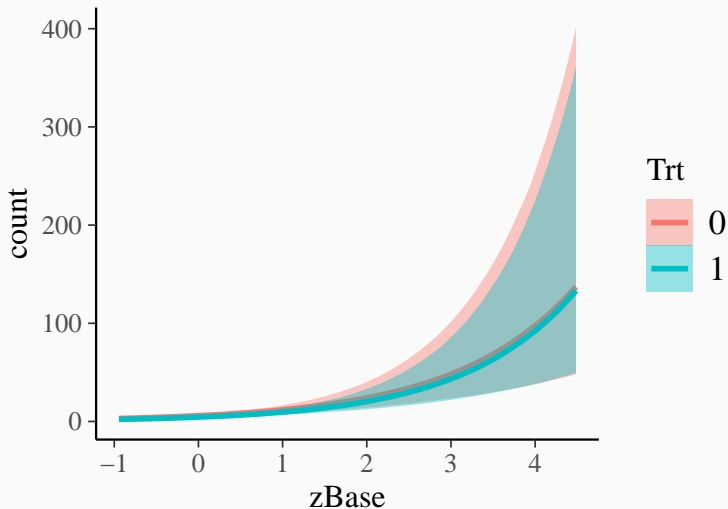
```
summary(fit_epi3)

## Family: poisson
## Links: mu = log
## Formula: count ~ zAge + zBase * Trt + (1 | patient)
## Data: epilepsy (Number of observations: 236)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~patient (Number of levels: 59)
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.58      0.07    0.46    0.74 1.00      758    1618
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        1.76      0.12    1.52    1.99 1.00      765    1468
## zAge              0.09      0.09   -0.07    0.28 1.00      807    1171
## zBase             0.71      0.12    0.47    0.94 1.01      805    1281
## Trt1             -0.27      0.17   -0.60    0.07 1.01      673    1387
## zBase:Trt1        0.05      0.17   -0.27    0.38 1.00      868    1358
##
## Samples were drawn using sampling(NUTS). 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).
```



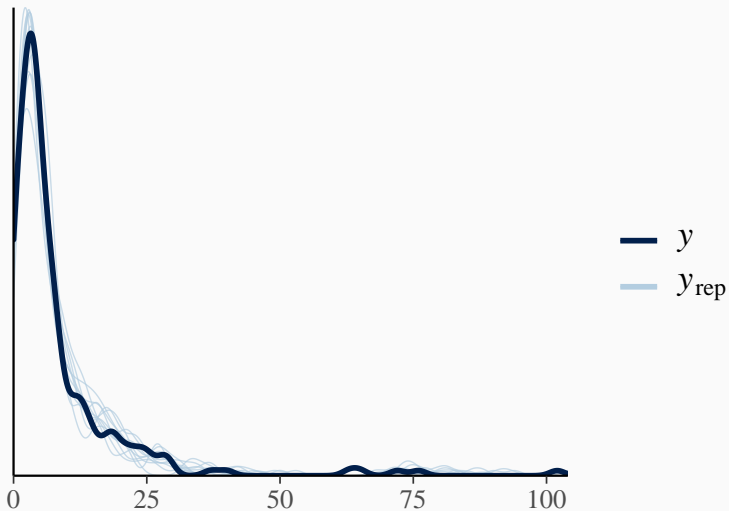
## Graphical Summary

```
conditional_effects(fit_epi3, "zBase:Trt")
```



# Posterior-Predictive Checks

```
pp_check(fit_epi3)
```



# Leave-One-Out Cross-Validation

```
loo(fit_epi3)

##
## Computed from 4000 by 236 log-likelihood matrix
##
##      Estimate   SE
## elpd_loo  -671.8 36.8
## p_loo      95.2 15.1
## looic      1343.7 73.6
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##
##      Count Pct.   Min. n_eff
## (-Inf, 0.5] (good)   209   88.6%   260
## (0.5, 0.7]  (ok)     16    6.8%   225
## (0.7, 1]    (bad)     10    4.2%    44
## (1, Inf)    (very bad) 1    0.4%     3
## See help('pareto-k-diagnostic') for details.
```

# Model Comparison

```
loo_compare(loo(fit_epi2), loo(fit_epi3))
```

```
##               elpd_diff se_diff  
## fit_epi3      0.0         0.0  
## fit_epi2 -202.9        63.1
```

## Epilepsy: Bayesian Model Building (4)

```
fit_epi4 <- brm(  
  count ~ zAge + zBase * Trt + (1 | patient),  
  data = epilepsy,  
  family = negbinomial("log"),  
  file = "models/fit_epi4"  
)  
  
fit_epi4 <- add_criterion(fit_epi4, "loo")
```

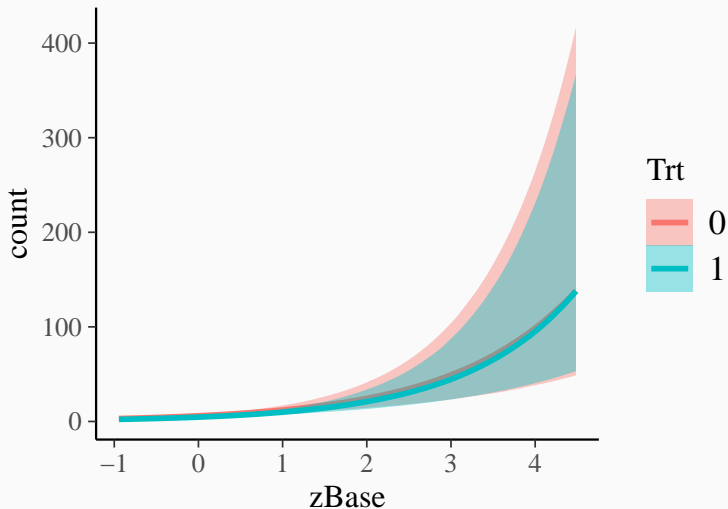
# Numerical Summary

```
summary(fit_epi4)

## Family: negbinomial
## Links: mu = log; shape = identity
## Formula: count ~ zAge + zBase * Trt + (1 | patient)
## Data: epilepsy (Number of observations: 236)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~patient (Number of levels: 59)
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.55     0.07    0.42    0.71 1.00    1254    1831
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        1.79     0.12     1.56     2.03 1.00    1743    2485
## zAge              0.09     0.09    -0.08     0.26 1.00    1767    2310
## zBase             0.70     0.12     0.47     0.94 1.00    1769    2011
## Trt1             -0.27     0.17    -0.59     0.06 1.00    1660    2241
## zBase:Trt1        0.06     0.16    -0.26     0.39 1.00    2045    2409
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape          7.41     1.77     4.70    11.65 1.00    3567    2972
##
## Samples were drawn using sampling(NUTS). 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).
```

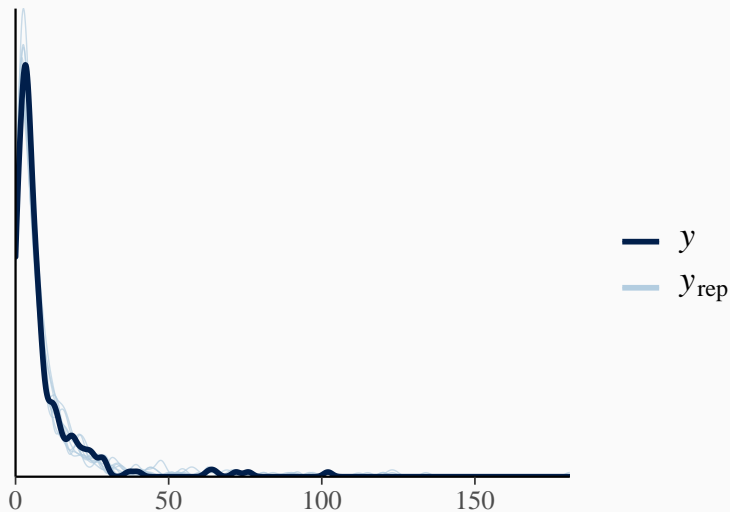
## Graphical Summary

```
conditional_effects(fit_epi4, "zBase:Trt")
```



# Posterior-Predictive Checks

```
pp_check(fit_epi4)
```





# Leave-One-Out Cross-Validation

```
loo(fit_epi4)

##
## Computed from 4000 by 236 log-likelihood matrix
##
##      Estimate   SE
## elpd_loo  -615.9 17.0
## p_loo      43.5  4.9
## looic      1231.9 33.9
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##               Count Pct.   Min. n_eff
## (-Inf, 0.5]  (good)   224  94.9%   977
## (0.5, 0.7]   (ok)      9   3.8%   371
## (0.7, 1]     (bad)      3   1.3%    35
## (1, Inf)     (very bad) 0   0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

# Model Comparison

```
loo_compare(loo(fit_epi2), loo(fit_epi3), loo(fit_epi4))
```

```
##           elpd_diff se_diff  
## fit_epi4      0.0       0.0  
## fit_epi3    -55.9      22.9  
## fit_epi2   -258.8      79.7
```

## Learn More

- Website: <https://paul-buerkner.github.io/>
- Email: [paul.buerkner@gmail.com](mailto:paul.buerkner@gmail.com)
- Twitter: @paulbuerkner

Learn more about brms:

- Github: <https://github.com/paul-buerkner/brms>
- Forums: <http://discourse.mc-stan.org/>
- Help within R: `help("brms")`
- Vignettes: `vignette(package = "brms")`

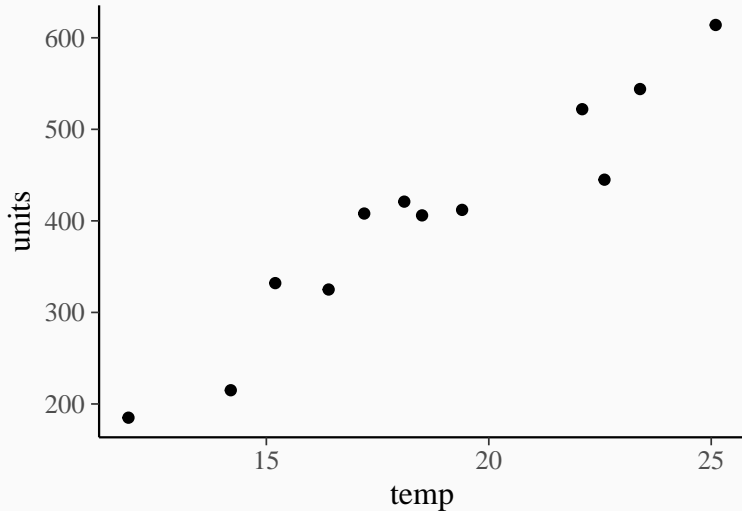
Learn more about Stan:

- Website: <http://mc-stan.org/>
- Forums: <http://discourse.mc-stan.org/>

# Appendix

## Beyond Inference and Prediction

# Bayesian Decision Theory

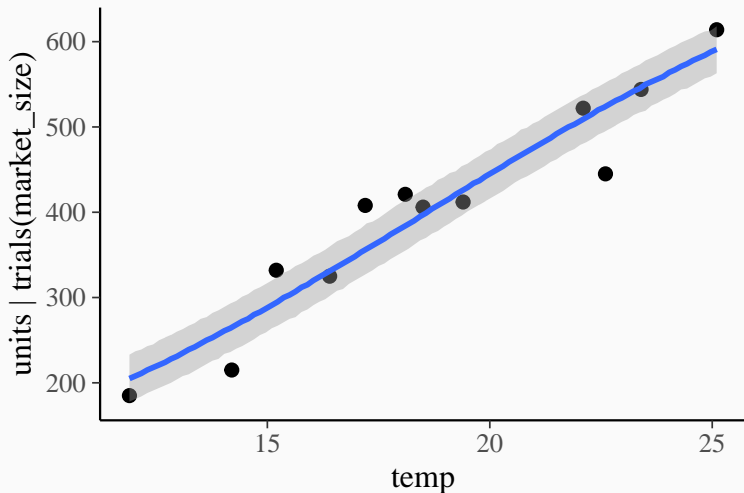


Thanks to Markus Gesmann!

# Predicting the Amount of Icecream Sold

Let's say the market size is 800 units of icecream

```
brm(units | trials(market_size) ~ temp, family = binomial())
```



## Deciding How Much Icecream to Buy

Our icecream truck costs 100€ per day

We buy each scoop of icecream for 1€ and sell it for 2€

Utility function:

$$U(x = x(T), b) = -100 - 1b + 2 \min(x, b)$$

We will optimize

$$\overline{U}(b) = \int U(x, b) \, d p(x)$$



## How Much Icecream To Buy?

We expect a temperature of 35 degrees

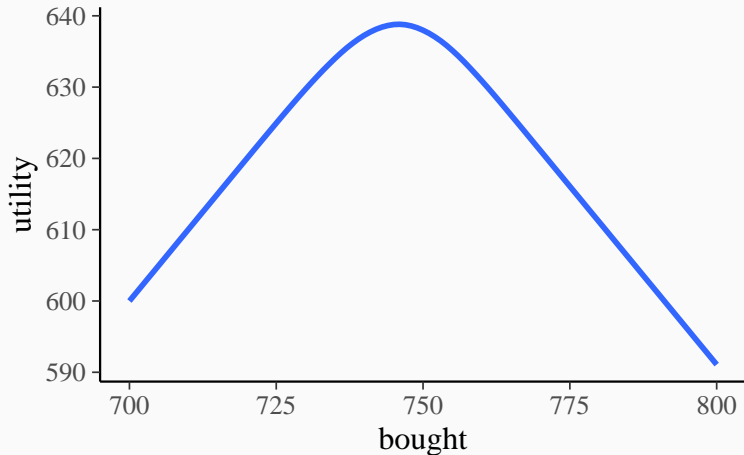
```
U <- function(units, bought) {  
  -100 - 1 * bought + 2 * pmin(units, bought)  
}
```

```
newdf <- data.frame(temp = 35, market_size = 800)  
pred <- posterior_predict(fit_ice1, newdata = newdf)  
bought <- 700:800
```

```
df <- bought %>%  
  map(~cbind(bought = ., utility = mean(U(pred, .)))) %>%  
  map(as_data_frame) %>%  
  bind_rows()
```

# How Much Icecream To Buy?

We expect a temperature of 35 degrees



Maximal utility of  $U = 638.8$  at 746 units bought

## Evaluating Prior Predictions

# The Bayes Factor

Marginal likelihood of model  $M$ :

$$p(y | M) = \int p(y | \theta, M) p(\theta | M) \mathrm{d} \theta$$

Bayes factor of models  $M_1$  vs.  $M_2$ :

$$\text{BF}_{12} = \frac{p(y | M_1)}{p(y | M_2)}$$

Evaluates **In-Sample Fit** and penalizes **Prior Complexity**

## Does recurrence time vary between women and men?

```
fit_kidney4 <- brm(  
  time | cens(censored) ~ age + sex,  
  family = Gamma("log"),  
  prior = prior(normal(0, 0.5), coef = "sexfemale"),  
  save_all_pars = TRUE,  
  ...  
)
```

```
fit_kidney5 <- brm(  
  time | cens(censored) ~ age,  
  family = Gamma("log"),  
  save_all_pars = TRUE,  
  ...  
)
```

## The Bayes Factor: Illustration

Does recurrence time vary between women and men?

```
bf54 <- bayes_factor(fit_kidney5, fit_kidney4)
```

Testing  $M_1 : b_{\text{sex}} = 0$  vs.  $M_2 : b_{\text{sex}} \neq 0$  reveals  $\text{BF}_{54} = 0.13$

