Bayesian Simple Regression

Rob Cribbie

15/11/2021

Open a Dataset - Real Data

Online Intervention for Perfectionism

```
dat<-read.csv(file.choose())
names(dat)
```

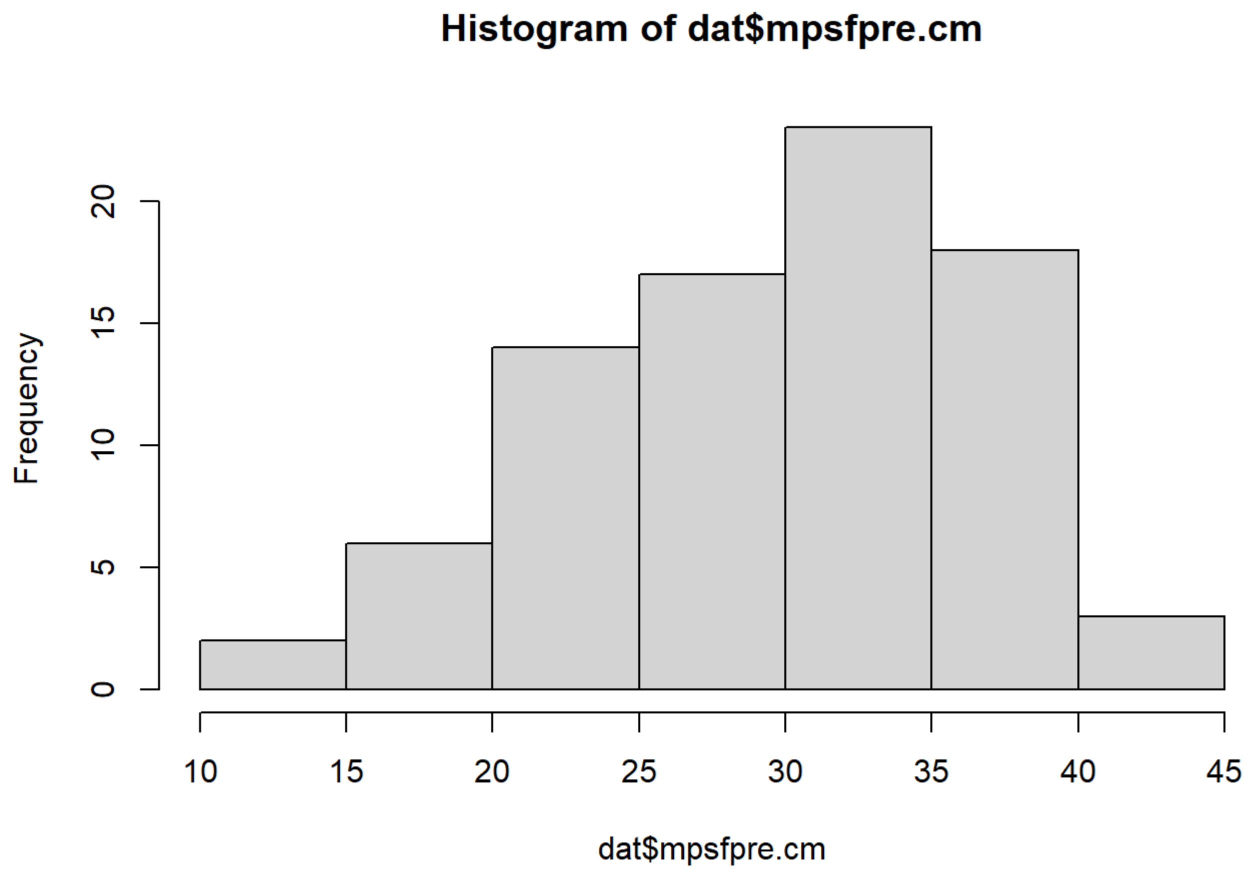
```
##  [1] "group"          "mpshfpre.sop"   "mpshfpre.spp"   "pcipre.total"
##  [5] "baipre.total"   "cesdpre.total"  "mpsfpre.cm"     "mpshfpost.sop"
##  [9] "mpshfpost.spp"  "pcipost.total"  "baipost.total"  "cesdpost.total"
## [13] "mpsfpost.cm"    "atqpre.total"   "atqpost.total"  "mpshfpre.oop"
## [17] "mpshfpost.oop"
```

We will be using a couple pretest variables

Predictor:

mpsfpre.cm (concern over mistakes)

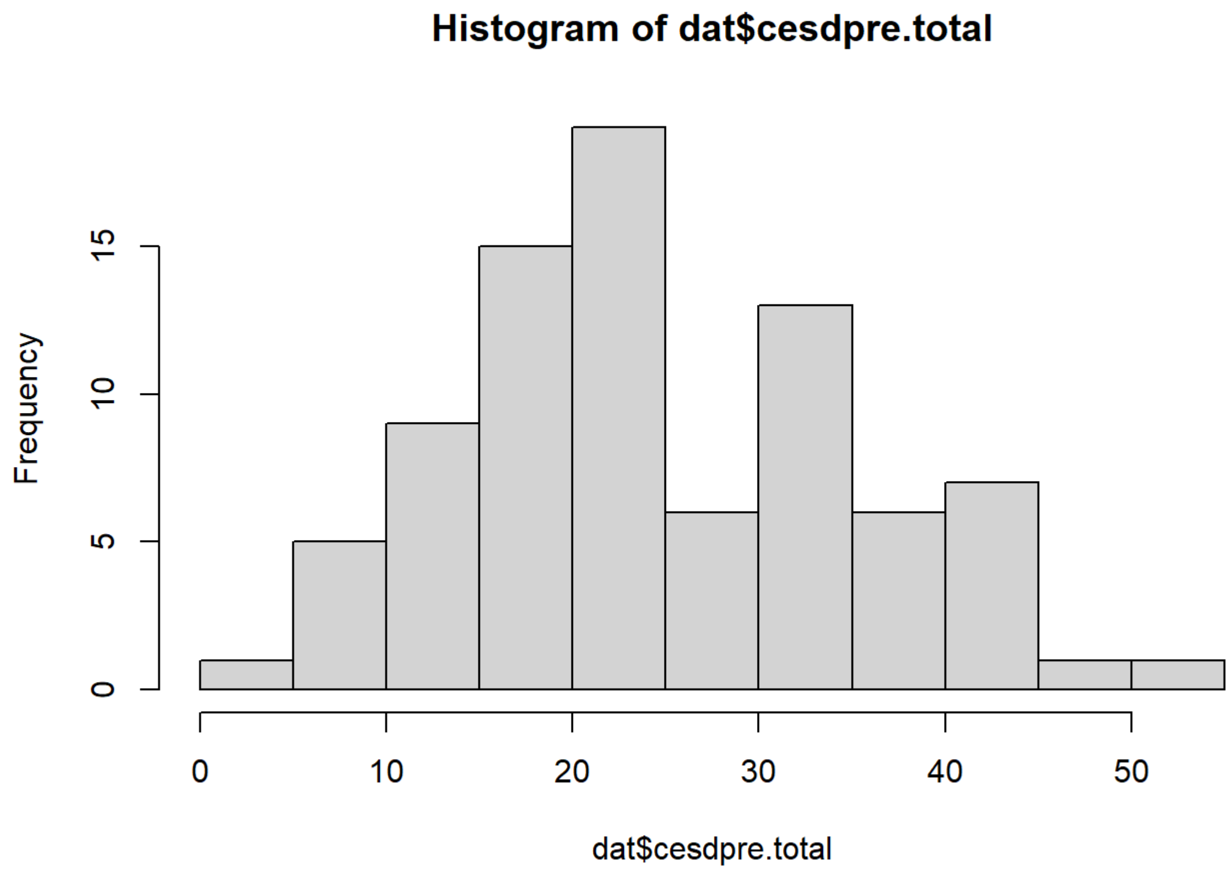
```
hist(dat$mpsfpre.cm)
```



Outcome:

cesdpre (depression)

```
hist(dat$cesdpre.total)
```



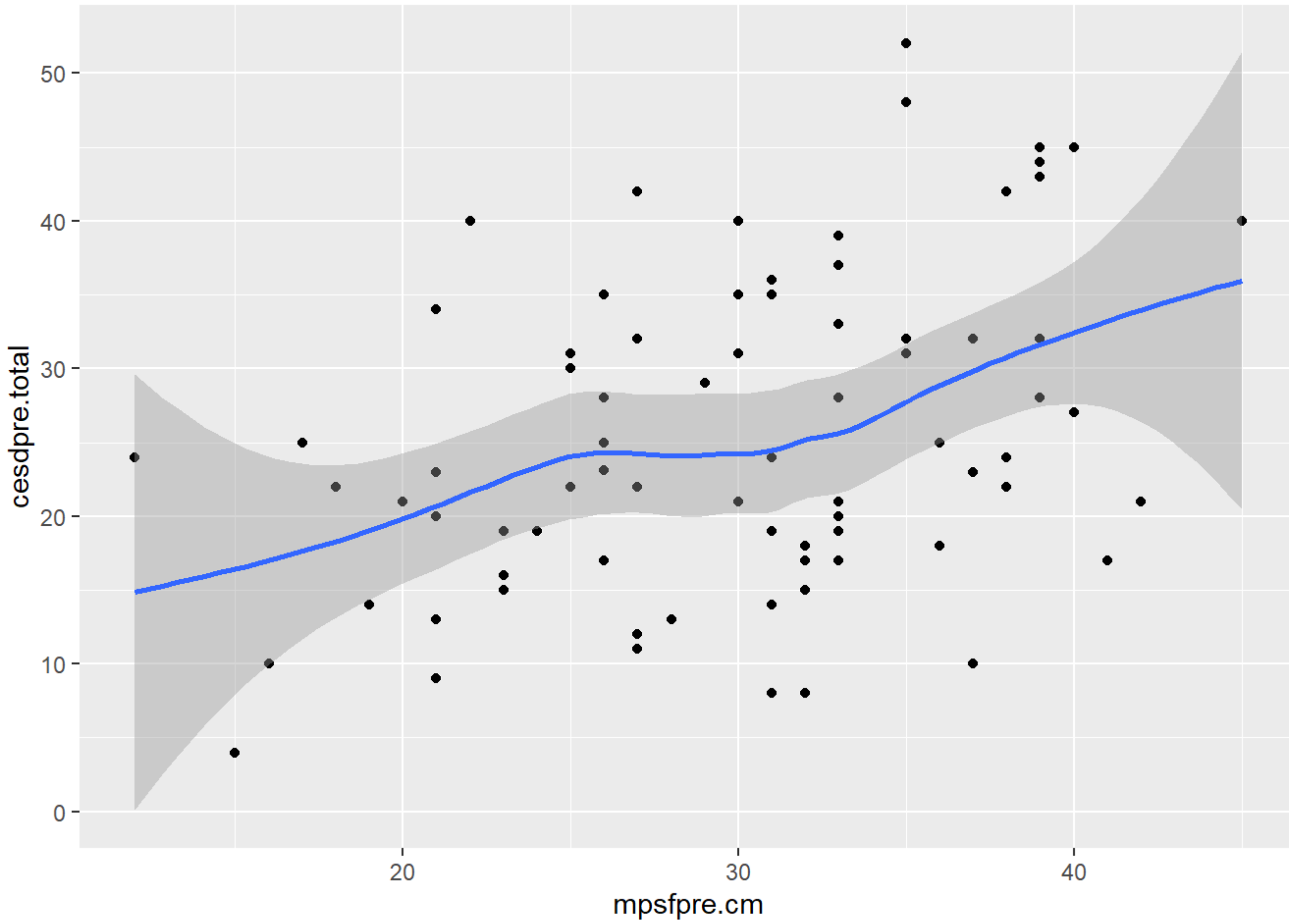
Our interest is in predicting depression from concern over mistakes

```
cor(dat$cesdpre.total,dat$mpsfpre.cm)
```

```
## [1] 0.4026916
```

```
library(ggplot2)
ggplot(dat,aes(x=mpsfpre.cm, y=cesdpre.total)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Traditional Regression Model

```
m<-lm(cesdpre.total ~ mpsfpre.cm, data=dat)
summary(m)
```

```
##
## Call:
## lm(formula = cesdpre.total ~ mpsfpre.cm, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.727  -7.058  -1.664   8.442  23.485
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.2935     4.7217   1.545 0.126320
## mpsfpre.cm    0.6063     0.1531   3.959 0.000161 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.936 on 81 degrees of freedom
## Multiple R-squared:  0.1622, Adjusted R-squared:  0.1518
## F-statistic: 15.68 on 1 and 81 DF,  p-value: 0.0001605
```

```
confint(m)
```

```
##                  2.5 %     97.5 %
## (Intercept) -2.1011402 16.6881038
## mpsfpre.cm   0.3016348  0.9110066
```

Interpretation

p-value: The probability of observing a test statistic this extreme, or more extreme, if Ho: b*=0 is true, is .00016

Confidence Interval: If we were to repeat the study many times, 95% of the CIs are expected to contain b*

Bayesian Regression (all defaults)

4 chains, 2000 samples per chain

Burn-in (warm-up): 1st 1000 samples of each chain

```
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.21.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defa
ults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
mb<-stan_glm(cesdpre.total ~ mpsfpre.cm,
             data=dat)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.049 seconds (Warm-up)
## Chain 1:                0.051 seconds (Sampling)
## Chain 1:                0.1 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.056 seconds (Warm-up)
## Chain 2:                0.051 seconds (Sampling)
## Chain 2:                0.107 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
```

```
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 0.053 seconds (Warm-up)
## Chain 3:                0.047 seconds (Sampling)
## Chain 3:                0.1 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 0.053 seconds (Warm-up)
## Chain 4:                0.051 seconds (Sampling)
## Chain 4:                0.104 seconds (Total)
## Chain 4: 
```

Equivalent Model (specifying number of chains and iterations):

```
mb<-stan_glm(cesdpre.total ~ mpsfpre.cm,
             chains=4, iter=2000,
             data=dat)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.067 seconds (Warm-up)
## Chain 1:                0.056 seconds (Sampling)
## Chain 1:                0.123 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.053 seconds (Warm-up)
## Chain 2:                0.062 seconds (Sampling)
## Chain 2:                0.115 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
```

```
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 0.055 seconds (Warm-up)
## Chain 3:                0.052 seconds (Sampling)
## Chain 3:                0.107 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 0.056 seconds (Warm-up)
## Chain 4:                0.053 seconds (Sampling)
## Chain 4:                0.109 seconds (Total)
## Chain 4: 
```

Priors

What priors are being used?

```
prior_summary(mb)
```

```
## Priors for model 'mb'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 25, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 25, scale = 27)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = 0, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 0, scale = 3.8)
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.093)
## ------
## See help('prior_summary.stanreg') for more details
```

Intercept Prior Calculation

Mean

```
mean(dat$cesdpre.total)
```

```
## [1] 25.4831
```

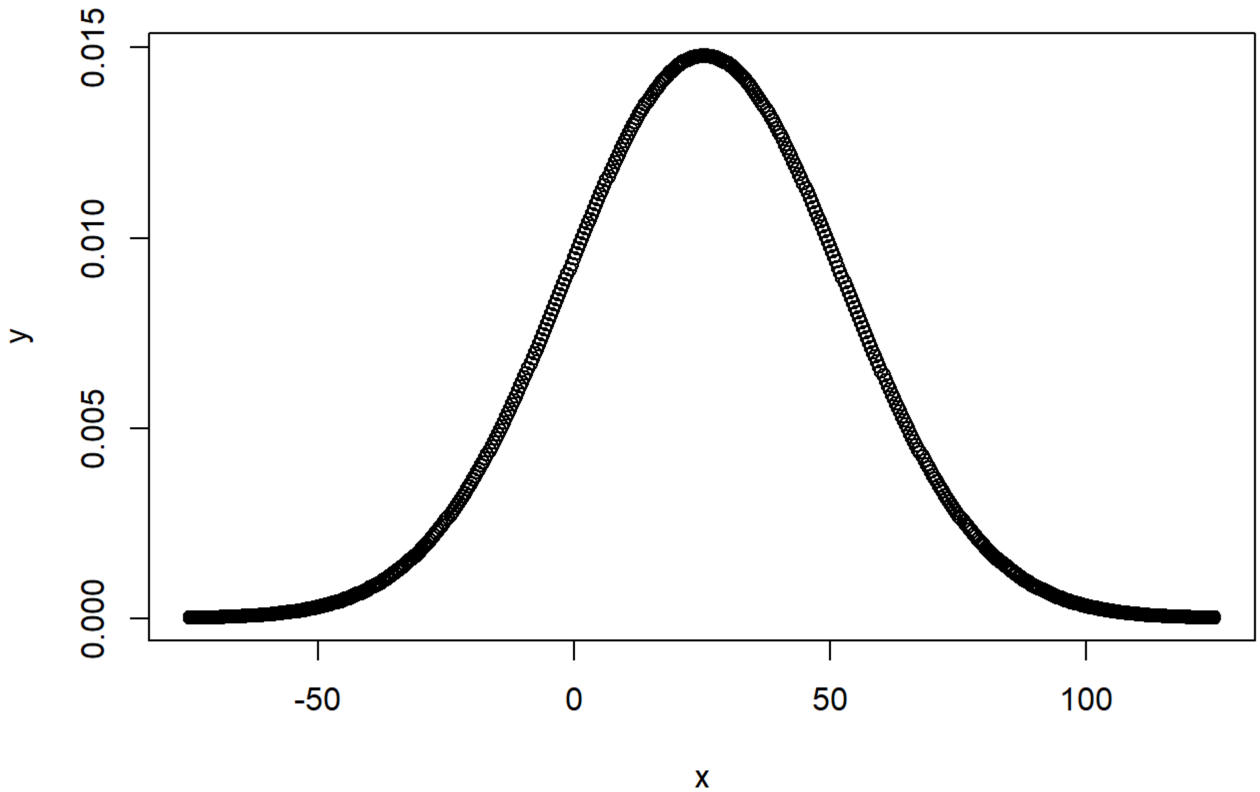
SD

```
sd(dat$cesdpre.total)*2.5
```

```
## [1] 26.97143
```

Plot the intercept prior

```
x<-seq(-75,125,length=500)
y<-dnorm(x, mean=mean(dat$cesdpre.total),
         sd=sd(dat$cesdpre.total)*2.5)
plot(x,y)
```



Regression Coefficient Prior Calculation

Mean (mean = 0)

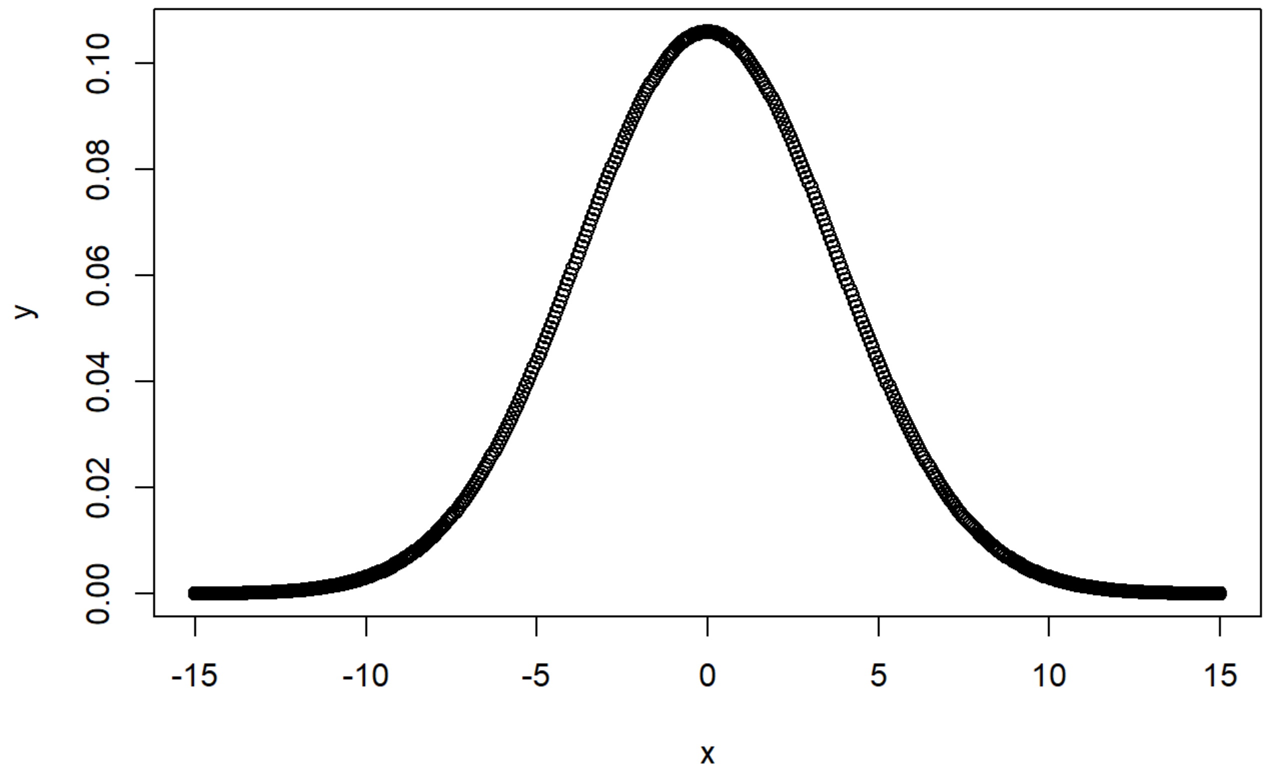
SD

```
sd(dat$cesdpre.total)/sd(dat$mpsfpre.cm)*2.5
```

```
## [1] 3.764175
```

Plot the regression coefficient prior

```
x<-seq(-15,15,length=500)
y<-dnorm(x, mean=0,
     sd=sd(dat$cesdpre.total)/sd(dat$mpsfpre.cm)*2.5)
plot(x,y)
```



Sigma Prior (SD of the residuals)

Exponential Rate

```
1/sd(dat$cesdpre.total)
```

```
## [1] 0.09269068
```

MCMC Progress/Convergence

Trace plot for chains by parameter

```
library(bayesplot)
```

```
## This is bayesplot version 1.8.1
```

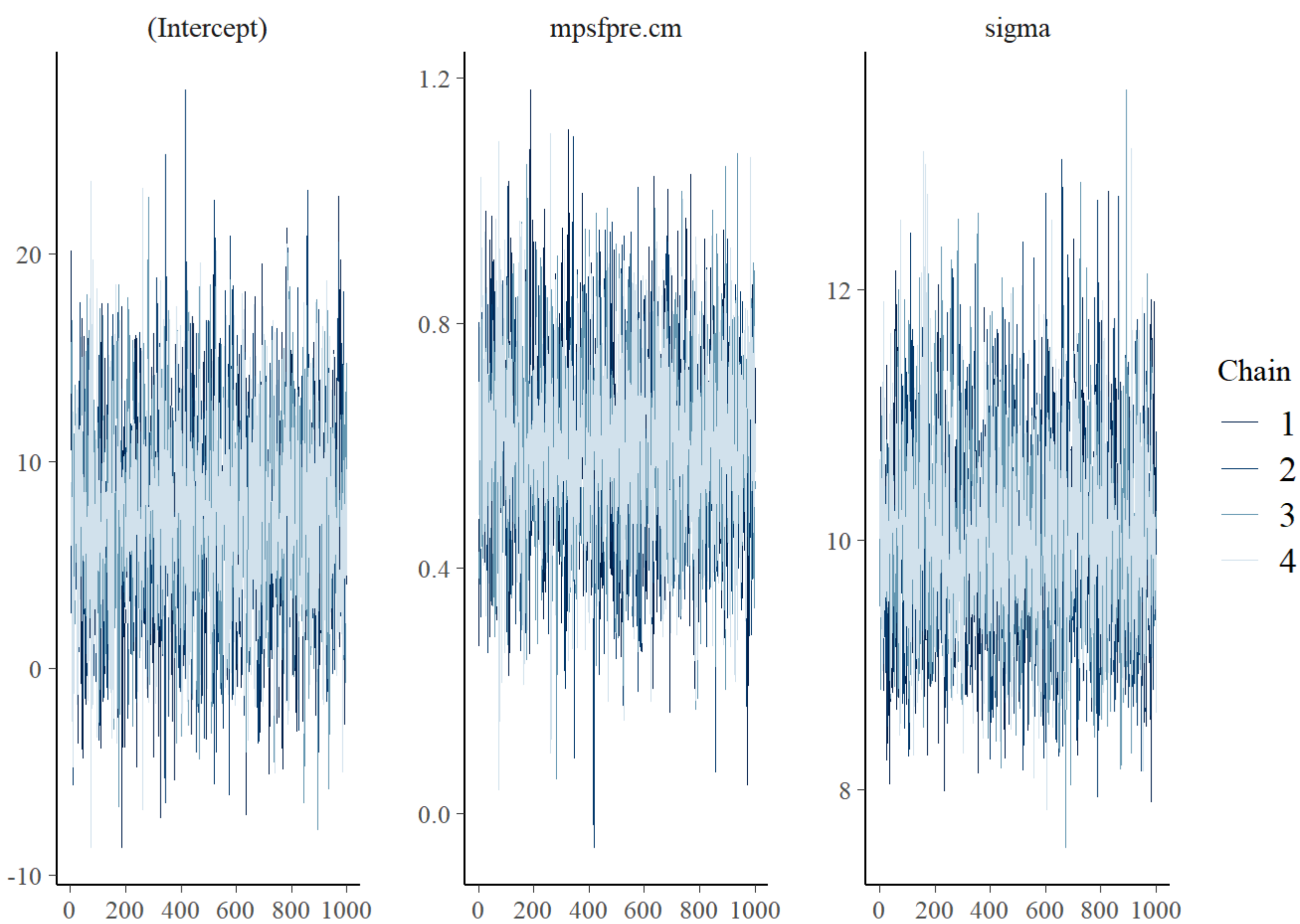
```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

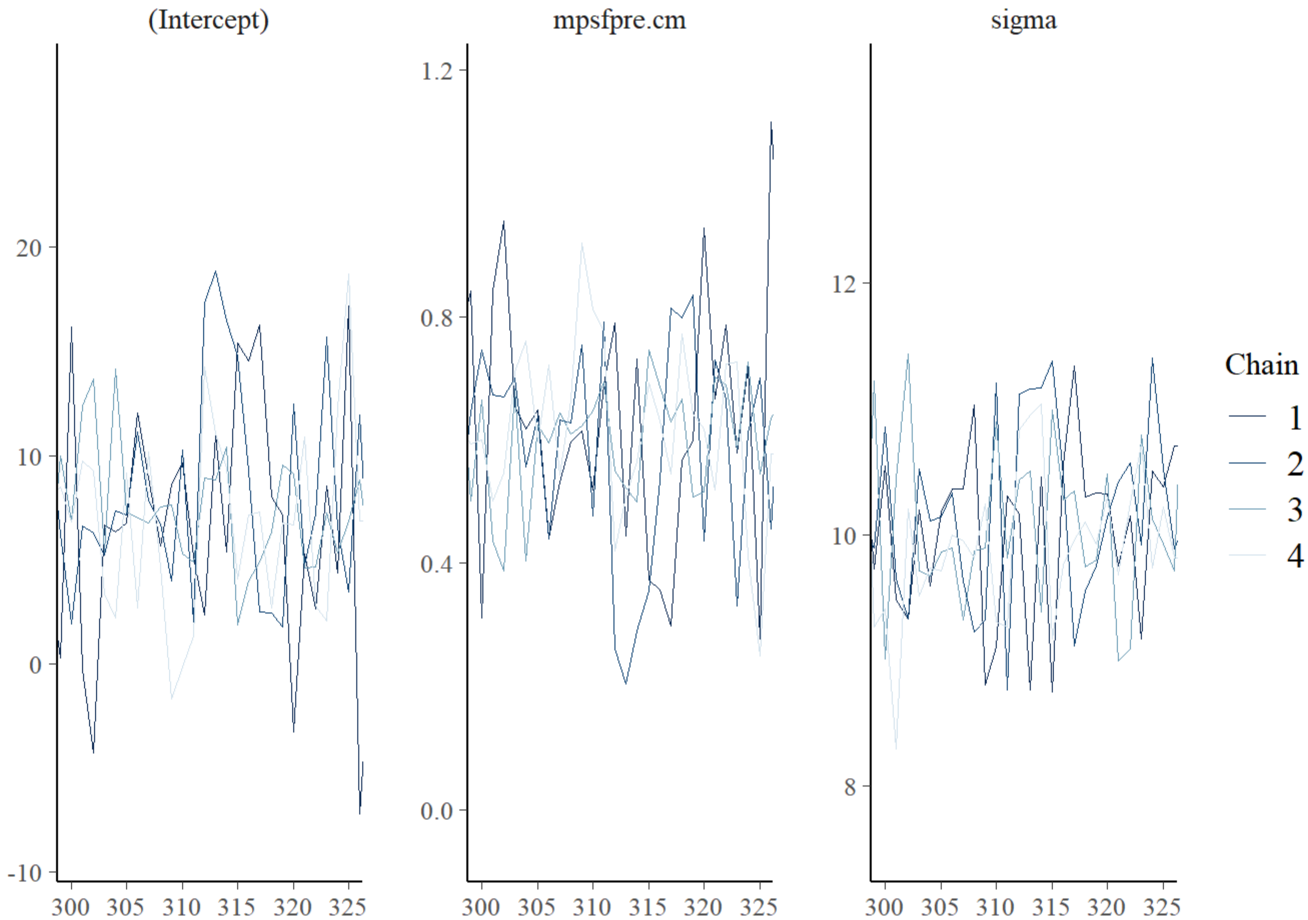
```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
mcmc_trace(mb)
```

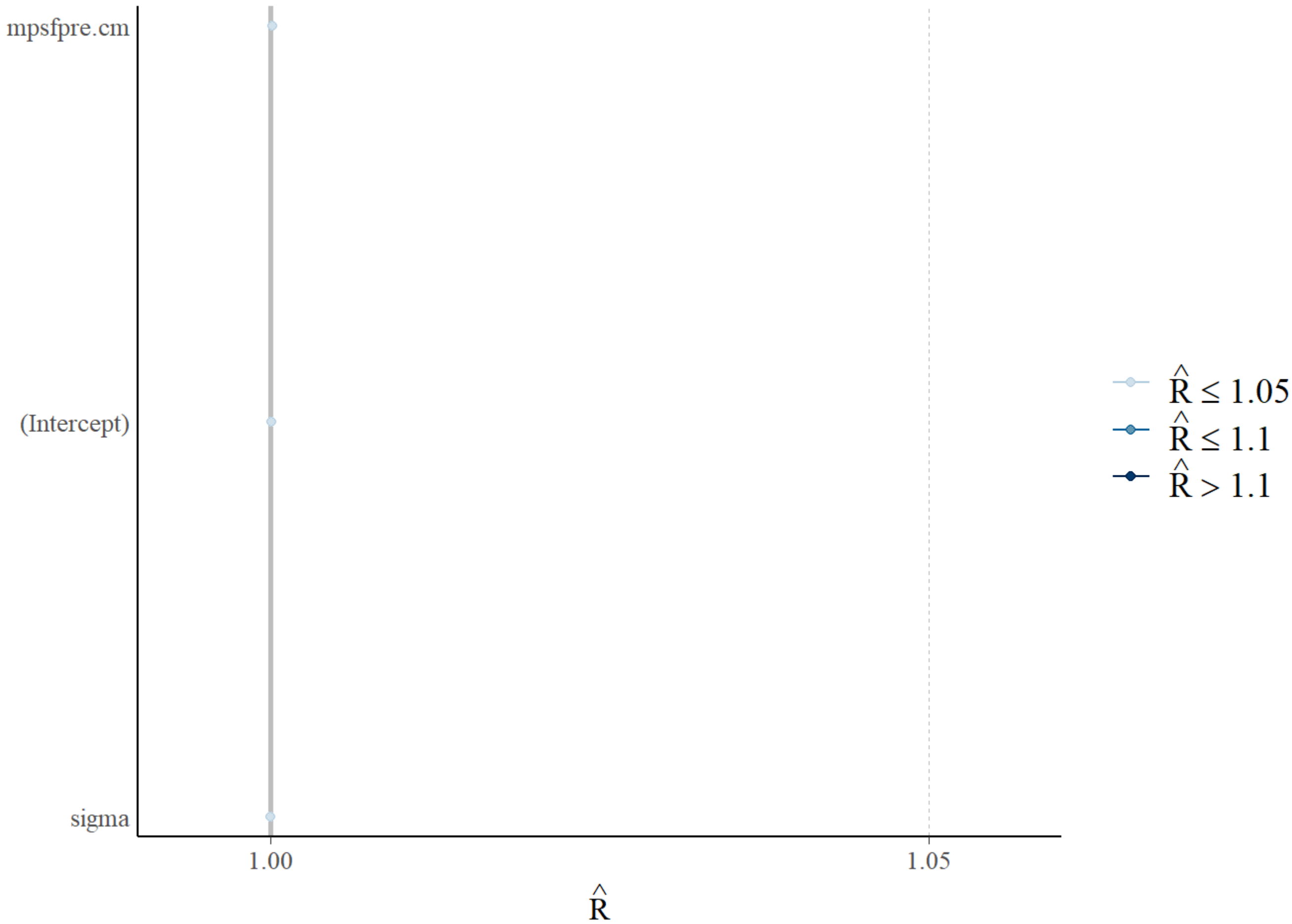



```
mcmc_trace(mb,window=c(300,325))
```



Plot of rhat: Comparison of between and within chain variances)

```
mcmc_rhat(rhat(mb)) + yaxis_text(hjust = 1)
```



mean_ppd, n_eff, mcse

```
summary(mb)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      cesdpre.total ~ mpsfpre.cm
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 83
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  7.4    4.8  1.2   7.4  13.6
## mpsfpre.cm   0.6    0.2  0.4   0.6   0.8
## sigma       10.0    0.8  9.0  10.0  11.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 25.5    1.5 23.6  25.5  27.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable
(for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.1  1.0  3736
## mpsfpre.cm    0.0  1.0  3720
## sigma         0.0  1.0  3453
## mean_PPD      0.0  1.0  4015
## log-posterior 0.0  1.0  1753
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective
sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rh
at=1).
```

```
mean(dat$cesdpre.total)
```

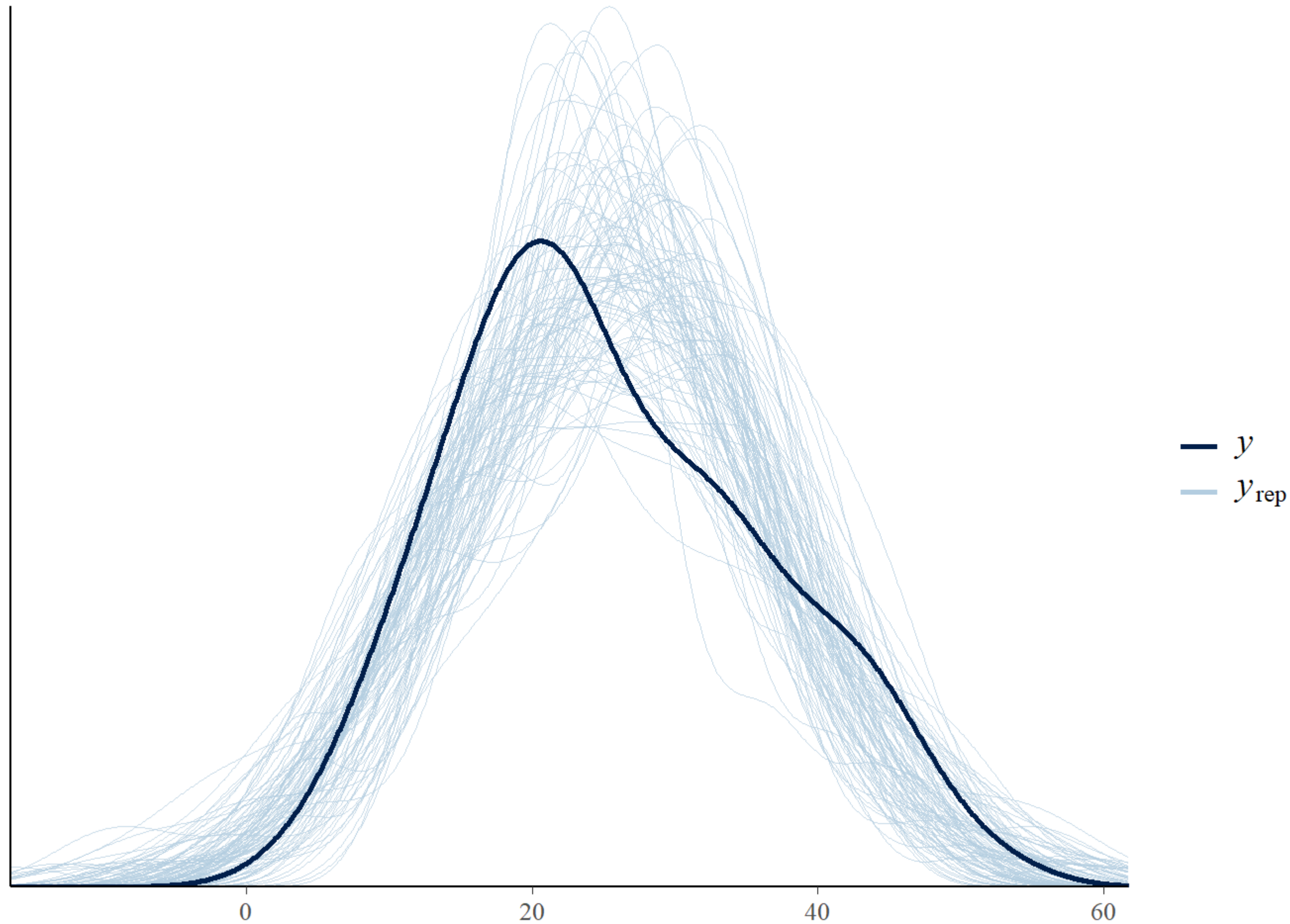
```
## [1] 25.4831
```

```
neff_ratio(mb)
```

```
## (Intercept)  mpsfpre.cm       sigma
##     0.93400     0.93000     0.86325
```

Posterior Predictive Check

```
pp_check(mb, nreps = 100)
```



Results Summary

```
summary(mb)
```

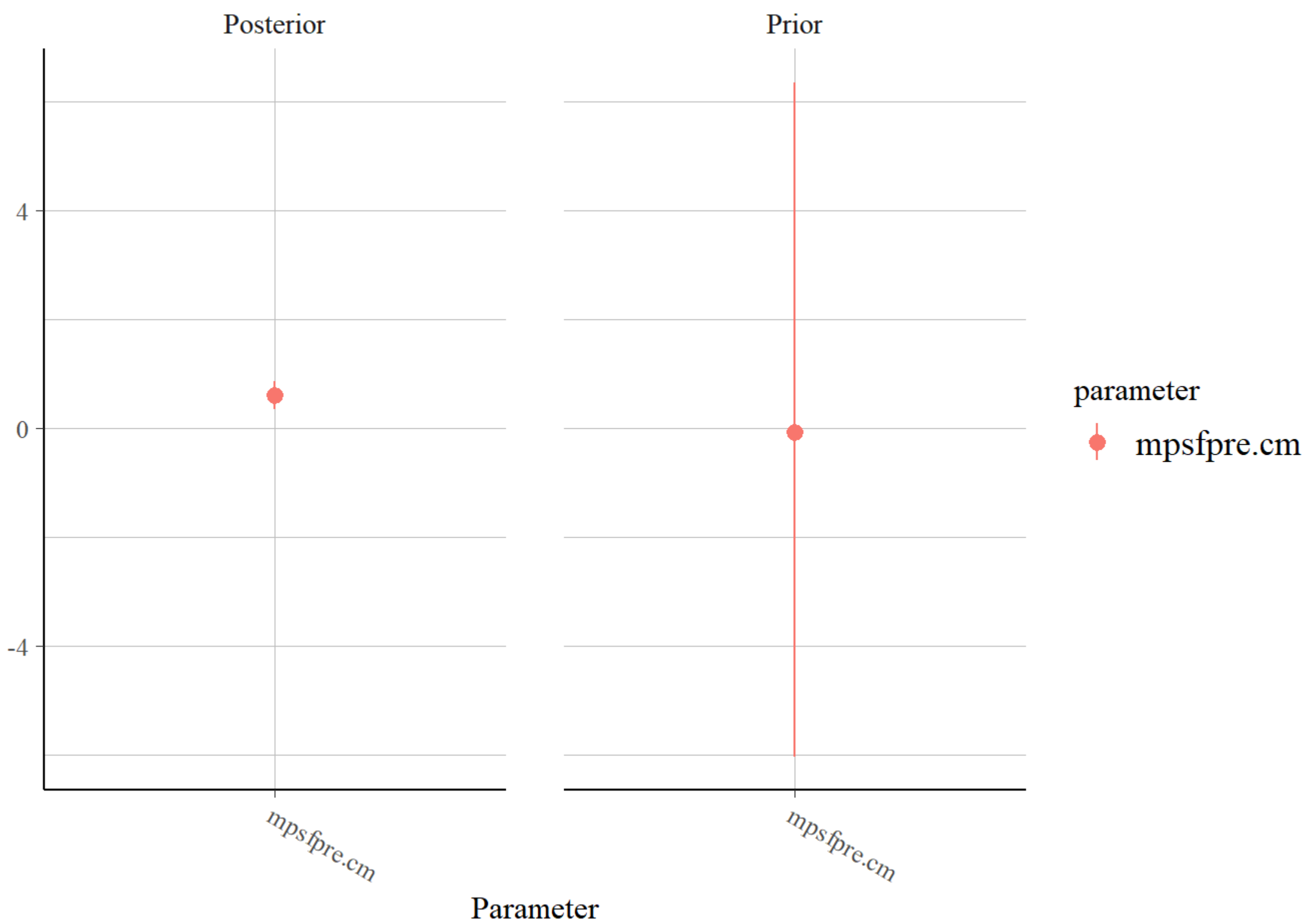
```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      cesdpre.total ~ mpsfpre.cm
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 83
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  7.4    4.8  1.2   7.4  13.6
## mpsfpre.cm   0.6    0.2  0.4   0.6   0.8
## sigma       10.0    0.8  9.0  10.0  11.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 25.5    1.5 23.6  25.5  27.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.1  1.0  3736
## mpsfpre.cm    0.0  1.0  3720
## sigma         0.0  1.0  3453
## mean_PPD      0.0  1.0  4015
## log-posterior 0.0  1.0  1753
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

```
coef(m)
```

```
## (Intercept)  mpsfpre.cm
##   7.2934818   0.6063207
```

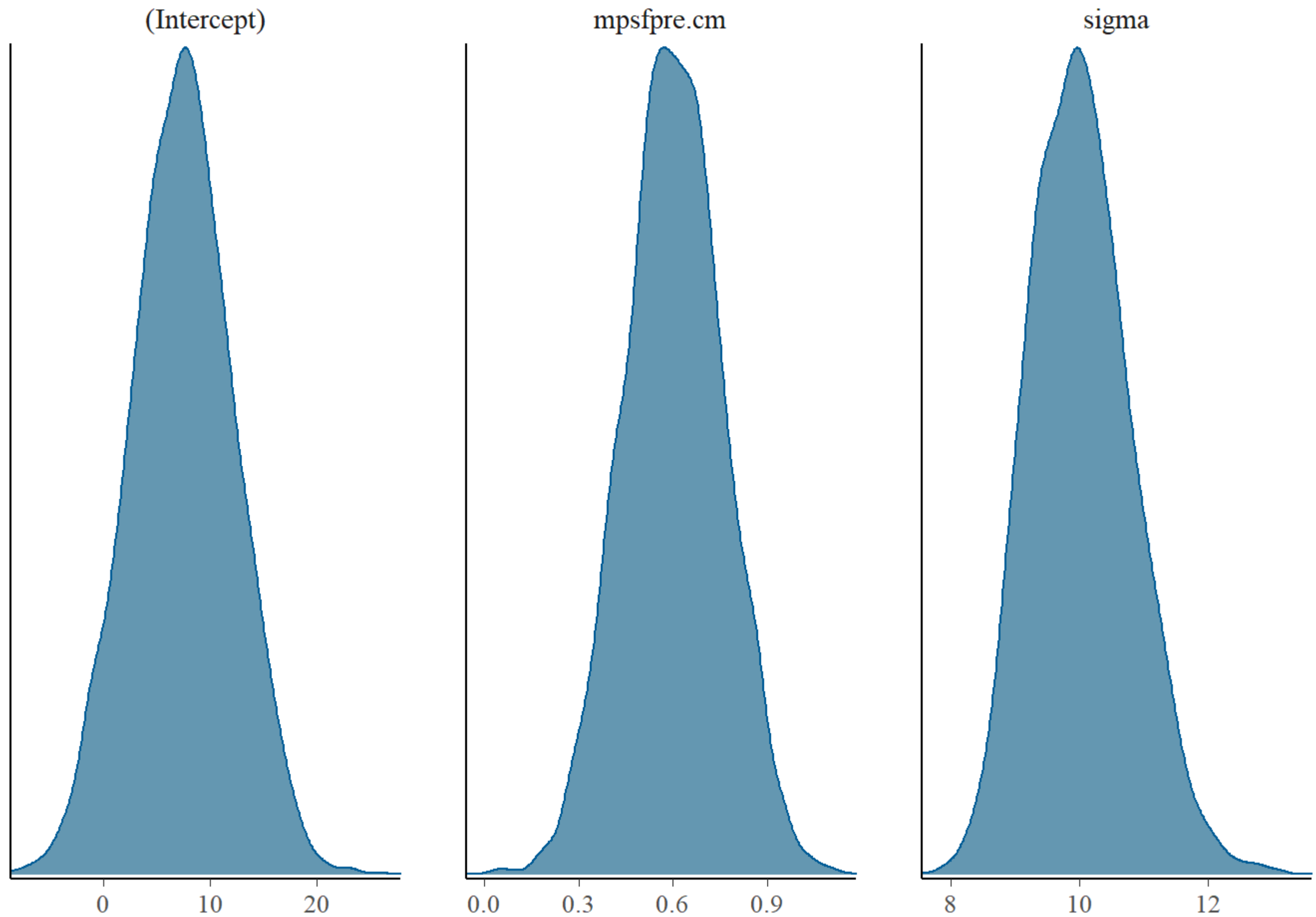
```
posterior_vs_prior(mb,pars = "beta")
```

```
##
## Drawing from prior...
```

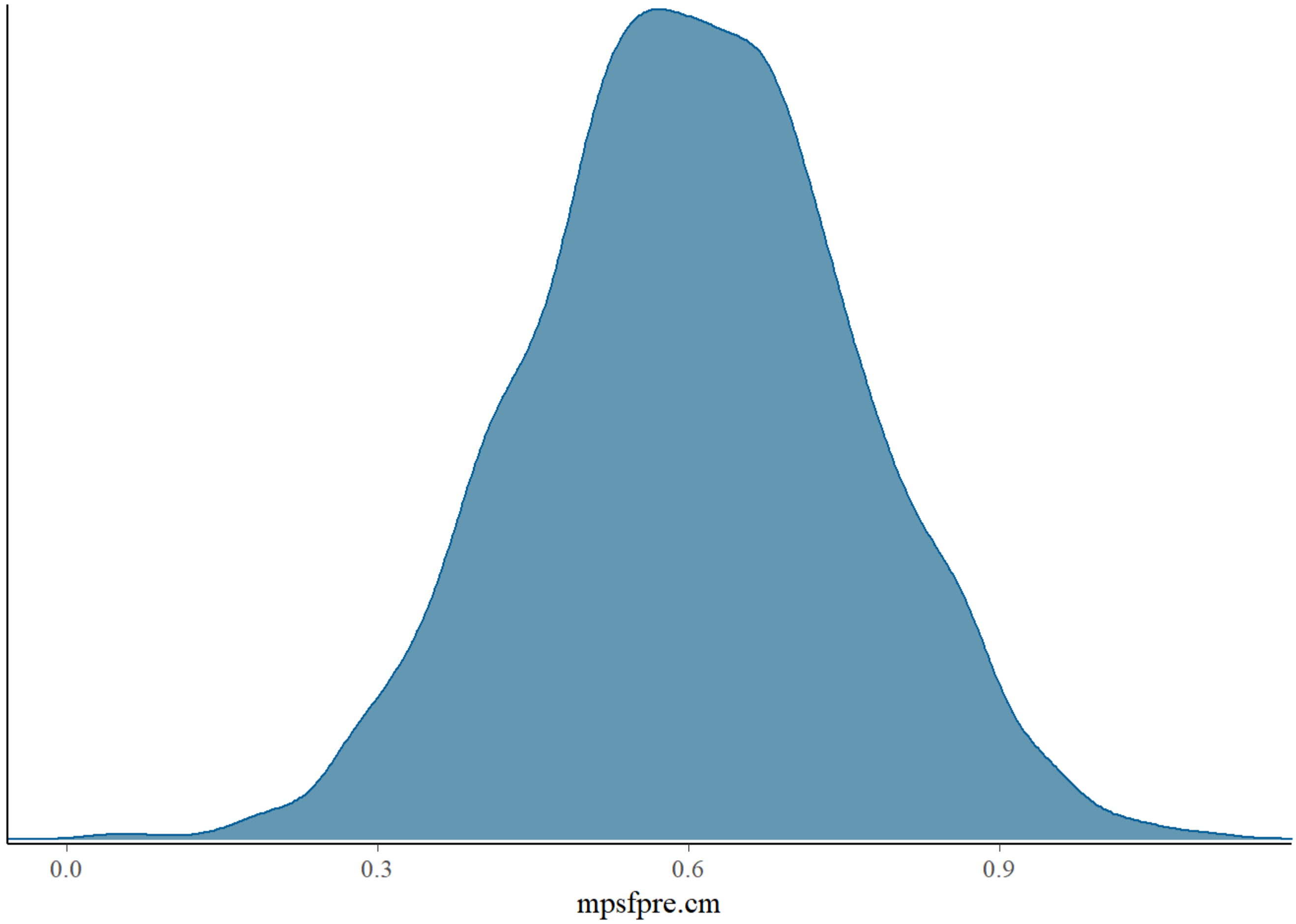


Distribution of the Parameters

```
library(bayesplot)
mcmc_dens(mb)
```




```
mcmc_dens(mb,pars=c("mpsfpre.cm"))
```




```
library(bayestestR)
describe_posterior(mb,
            parameters = "mpsfpre.cm",
            test="none", ci_method="HDI")
```

```
## Summary of Posterior Distribution
##
## Parameter  | Median |       95% CI |  Rhat |     ESS
## ----------------------------------------------------
## mpsfpre.cm |   0.60 | [0.29, 0.90] | 1.000 | 3720.00
```

CI Interpretation: There is a 95% chance that the true parameter falls between .31 and .92 (Much improved!)

Explore the posterior (for our reg coef)

```
post<-as.matrix(mb)
sum(post[,"mpsfpre.cm"]<0)/4000
```

```
## [1] 0.00025
```

Interpretation: There is 0% chance that the regression parameter falls below 0

Minimum posterior value for b

```
min(post[,"mpsfpre.cm"])
```

```
## [1] -0.05618353
```

Equivalence Test (Ho: b<=-*.3 | b*>.3)

```
describe_posterior(mb, rope_range=c(-.3,.3),
               parameters = "mpsfpre.cm",
               test=c("rope","equitest"))
```

```
## Summary of Posterior Distribution
##
## Parameter  | Median |       95% CI |          ROPE | % in ROPE | Equivalence (ROPE) |  Rhat |
ESS
## ---------------------------------------------------------------------------------------------
--------
## mpsfpre.cm |   0.60 | [0.29, 0.90] | [-0.30, 0.30] |     0.45% |          Undecided | 1.000 |
3720.00
```

Bayes Factors (Effect Size?)

```
bayesfactor_parameters(mb)
```

```
## Sampling priors, please wait...
```

```
## Bayes Factor (Savage-Dickey density ratio)
##
## Parameter   |    BF
## -------------------
## (Intercept) | 0.128
## mpsfpre.cm  | 15.73
##
## * Evidence Against The Null: 0
```

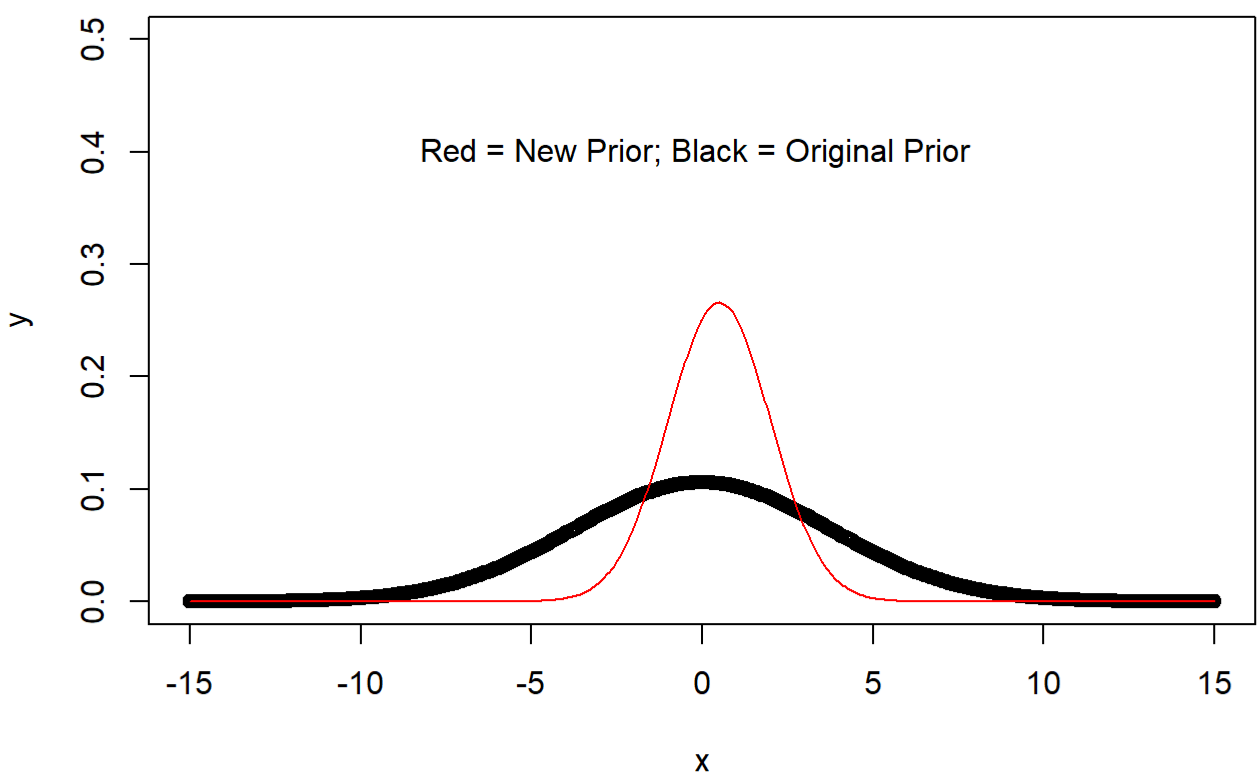
```
bayesfactor_parameters(mb, null=c(-.3,.3))
```

```
## Sampling priors, please wait...
```

```
## Bayes Factor (Null-Interval)
##
## Parameter    |     BF
## -------------------
## (Intercept) | 0.123
## mpsfpre.cm  |  2.44
##
## * Evidence Against The Null: [-0.300, 0.300]
```

Change Prior (More informative)

```
x<-seq(-15,15,length=500)
y<-dnorm(x, mean=0,
         sd=sd(dat$cesdpre.total)/sd(dat$mpsfpre.cm)*2.5)
plot(x,y,ylim=c(0,.5))
y1<-dnorm(x, mean=.5, sd=1.5)
lines(x,y1,col="red")
text(-.2,.4,"Red = New Prior; Black = Original Prior")
```



```
mb2<-stan_glm(cesdpre.total ~ mpsfpre.cm,
     prior = normal(.5, 1.5, autoscale=FALSE),
     data=dat)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.041 seconds (Warm-up)
## Chain 1:                0.045 seconds (Sampling)
## Chain 1:                0.086 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.051 seconds (Warm-up)
## Chain 2:                0.043 seconds (Sampling)
## Chain 2:                0.094 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
```

```
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.048 seconds (Warm-up)
## Chain 3:                0.048 seconds (Sampling)
## Chain 3:                0.096 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.048 seconds (Warm-up)
## Chain 4:                0.048 seconds (Sampling)
## Chain 4:                0.096 seconds (Total)
## Chain 4:
```

```
prior_summary(mb2)
```

```
## Priors for model 'mb2'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 25, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 25, scale = 27)
##
## Coefficients
##  ~ normal(location = 0.5, scale = 1.5)
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.093)
## ------
## See help('prior_summary.stanreg') for more details
```

```
describe_posterior(mb2,
       parameters = "mpsfpre.cm",
       test="none", ci_method="HDI")
```

```
## Summary of Posterior Distribution
##
## Parameter  | Median |       95% CI |  Rhat |     ESS
## ----------------------------------------------------
## mpsfpre.cm |   0.61 | [0.30, 0.91] | 1.000 | 4024.00
```