Mesocosm Results

Brianna Henry

September 27, 2017

library(brms)

## Warning: package 'brms' was built under R version 3.3.3

## Loading required package: Rcpp

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.3.3

## Loading 'brms' package (version 2.1.0). Useful instructions  
## can be found by typing help('brms'). A more detailed introduction  
## to the package is available through vignette('brms\_overview').  
## Run theme\_set(theme\_default()) to use the default bayesplot theme.

library(ggplot2)

In all data, treatment 1 is control, treatment 2 is atrazine, treatment 3 is selenium, and treatment 4 is the atrazine and selenium combined treatment.

Adult Biomass

Adult biomass data is from the biodivnew file and is initially in g/0.36m2/4 days.

urlfile<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/biodivnew.csv"  
biodivnew<-read.csv(urlfile)  
biodivnew$trt0f<-as.factor(biodivnew$trt0f)  
biodivnew$date0f<-as.factor(biodivnew$date0f)

Total Biomass

Model

get\_prior(totbio~date0f+trt0f+date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f6D28D2016:trt0f2   
## 4 b date0f6D28D2016:trt0f3   
## 5 b date0f6D28D2016:trt0f4   
## 6 b date0f7D12D2016   
## 7 b date0f7D12D2016:trt0f2   
## 8 b date0f7D12D2016:trt0f3   
## 9 b date0f7D12D2016:trt0f4   
## 10 b date0f7D18D2016   
## 11 b date0f7D18D2016:trt0f2   
## 12 b date0f7D18D2016:trt0f3   
## 13 b date0f7D18D2016:trt0f4   
## 14 b date0f7D26D2016   
## 15 b date0f7D26D2016:trt0f2   
## 16 b date0f7D26D2016:trt0f3   
## 17 b date0f7D26D2016:trt0f4   
## 18 b date0f7D4D2016   
## 19 b date0f7D4D2016:trt0f2   
## 20 b date0f7D4D2016:trt0f3   
## 21 b date0f7D4D2016:trt0f4   
## 22 b date0f8D2D2016   
## 23 b date0f8D2D2016:trt0f2   
## 24 b date0f8D2D2016:trt0f3   
## 25 b date0f8D2D2016:trt0f4   
## 26 b trt0f2   
## 27 b trt0f3   
## 28 b trt0f4   
## 29 student\_t(3, -4, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## 33 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33

mbiobrm<-brm(totbio~date0f+trt0f+date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.854 seconds (Warm-up)  
## 5.978 seconds (Sampling)  
## 11.832 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.553 seconds (Warm-up)  
## 4.978 seconds (Sampling)  
## 10.531 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.07 seconds (Warm-up)  
## 5.801 seconds (Sampling)  
## 11.871 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.307 seconds (Warm-up)  
## 5.016 seconds (Sampling)  
## 11.323 seconds (Total)

Model Checks

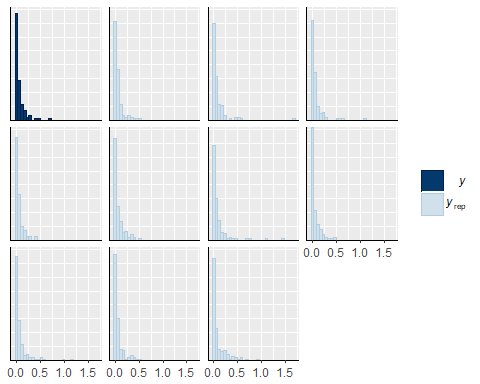
print(mbiobrm)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: totbio ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: biodivnew (Number of observations: 164)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.24 0.15 0.01 0.56 1388 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept -4.47 0.38 -5.19 -3.67 1637  
## date0f6D28D2016 0.88 0.55 -0.18 1.99 2318  
## date0f7D12D2016 0.03 0.54 -1.02 1.08 2334  
## date0f7D18D2016 2.16 0.54 1.02 3.19 2154  
## date0f7D26D2016 2.07 0.52 1.06 3.09 2267  
## date0f7D4D2016 1.59 0.52 0.55 2.64 2040  
## date0f8D2D2016 1.99 0.52 0.98 3.02 2042  
## trt0f2 -0.14 0.51 -1.12 0.89 1752  
## trt0f3 0.05 0.51 -0.93 1.05 1852  
## trt0f4 -0.11 0.51 -1.14 0.89 1930  
## date0f6D28D2016:trt0f2 0.07 0.78 -1.43 1.57 2643  
## date0f7D12D2016:trt0f2 0.02 0.72 -1.43 1.42 2469  
## date0f7D18D2016:trt0f2 0.53 0.76 -0.97 2.05 2501  
## date0f7D26D2016:trt0f2 0.08 0.74 -1.30 1.56 2418  
## date0f7D4D2016:trt0f2 -0.51 0.72 -1.91 0.88 2311  
## date0f8D2D2016:trt0f2 0.38 0.74 -1.03 1.84 2548  
## date0f6D28D2016:trt0f3 -0.03 0.76 -1.50 1.52 2624  
## date0f7D12D2016:trt0f3 0.29 0.74 -1.16 1.78 2566  
## date0f7D18D2016:trt0f3 -1.41 0.75 -2.81 0.07 2606  
## date0f7D26D2016:trt0f3 0.91 0.75 -0.59 2.39 2776  
## date0f7D4D2016:trt0f3 0.13 0.74 -1.28 1.59 2881  
## date0f8D2D2016:trt0f3 -0.23 0.74 -1.72 1.25 2657  
## date0f6D28D2016:trt0f4 0.18 0.76 -1.26 1.70 2547  
## date0f7D12D2016:trt0f4 -0.24 0.75 -1.70 1.24 2561  
## date0f7D18D2016:trt0f4 -0.01 0.76 -1.47 1.53 3000  
## date0f7D26D2016:trt0f4 0.43 0.73 -1.01 1.82 2767  
## date0f7D4D2016:trt0f4 -0.24 0.74 -1.67 1.21 2356  
## date0f8D2D2016:trt0f4 0.32 0.74 -1.15 1.84 2497  
## Rhat  
## Intercept 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016:trt0f2 1.00  
## date0f7D12D2016:trt0f2 1.00  
## date0f7D18D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f7D4D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f6D28D2016:trt0f3 1.00  
## date0f7D12D2016:trt0f3 1.00  
## date0f7D18D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f7D4D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f6D28D2016:trt0f4 1.00  
## date0f7D12D2016:trt0f4 1.00  
## date0f7D18D2016:trt0f4 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f7D4D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 0.87 0.09 0.70 1.05 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(mbiobrm,type="hist")

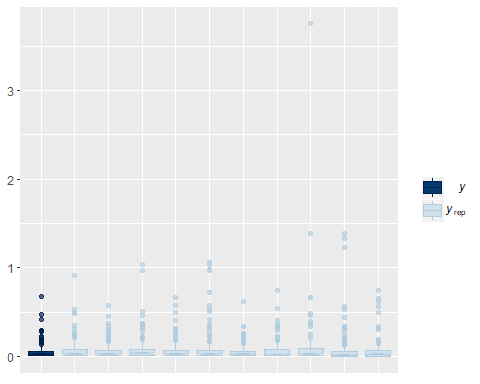
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(mbiobrm,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Chironomid Biomass

Model

get\_prior(biochiro~trt0f+date0f+date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f7D12D2016   
## 4 b date0f7D18D2016   
## 5 b date0f7D26D2016   
## 6 b date0f7D4D2016   
## 7 b date0f8D2D2016   
## 8 b trt0f2   
## 9 b trt0f2:date0f6D28D2016   
## 10 b trt0f2:date0f7D12D2016   
## 11 b trt0f2:date0f7D18D2016   
## 12 b trt0f2:date0f7D26D2016   
## 13 b trt0f2:date0f7D4D2016   
## 14 b trt0f2:date0f8D2D2016   
## 15 b trt0f3   
## 16 b trt0f3:date0f6D28D2016   
## 17 b trt0f3:date0f7D12D2016   
## 18 b trt0f3:date0f7D18D2016   
## 19 b trt0f3:date0f7D26D2016   
## 20 b trt0f3:date0f7D4D2016   
## 21 b trt0f3:date0f8D2D2016   
## 22 b trt0f4   
## 23 b trt0f4:date0f6D28D2016   
## 24 b trt0f4:date0f7D12D2016   
## 25 b trt0f4:date0f7D18D2016   
## 26 b trt0f4:date0f7D26D2016   
## 27 b trt0f4:date0f7D4D2016   
## 28 b trt0f4:date0f8D2D2016   
## 29 student\_t(3, -5, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## 33 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33

chirobio<-brm(biochiro~date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.33 seconds (Warm-up)  
## 6.419 seconds (Sampling)  
## 12.749 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.824 seconds (Warm-up)  
## 5.704 seconds (Sampling)  
## 11.528 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.098 seconds (Warm-up)  
## 7.96 seconds (Sampling)  
## 14.058 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.838 seconds (Warm-up)  
## 4.685 seconds (Sampling)  
## 10.523 seconds (Total)

Model Checks

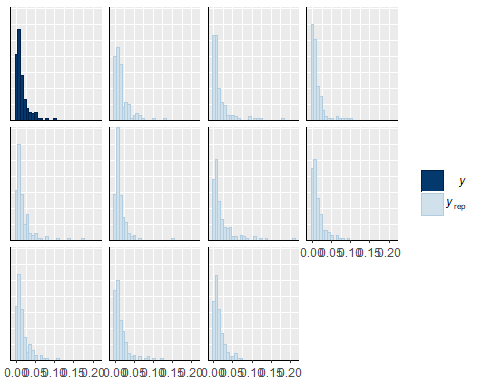
print(chirobio)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: biochiro ~ date0f \* trt0f + (1 | tank)   
## Data: biodivnew (Number of observations: 164)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.33 0.12 0.07 0.59 1010 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept -4.77 0.32 -5.38 -4.11 1213  
## date0f6D28D2016 1.05 0.43 0.20 1.91 1771  
## date0f7D12D2016 0.19 0.42 -0.63 1.01 1338  
## date0f7D18D2016 -0.63 0.41 -1.44 0.19 1412  
## date0f7D26D2016 0.04 0.42 -0.79 0.88 1633  
## date0f7D4D2016 1.63 0.40 0.83 2.41 1598  
## date0f8D2D2016 0.23 0.41 -0.54 1.05 1686  
## trt0f2 -0.01 0.43 -0.86 0.82 1518  
## trt0f3 0.19 0.43 -0.66 1.06 1345  
## trt0f4 0.04 0.43 -0.80 0.89 1488  
## date0f6D28D2016:trt0f2 -0.18 0.62 -1.43 1.05 2312  
## date0f7D12D2016:trt0f2 -0.02 0.59 -1.14 1.14 1763  
## date0f7D18D2016:trt0f2 -0.00 0.59 -1.12 1.21 1755  
## date0f7D26D2016:trt0f2 -1.14 0.58 -2.32 -0.02 1958  
## date0f7D4D2016:trt0f2 -0.51 0.57 -1.58 0.62 2036  
## date0f8D2D2016:trt0f2 -0.12 0.57 -1.26 0.98 2122  
## date0f6D28D2016:trt0f3 -0.24 0.59 -1.43 0.88 2049  
## date0f7D12D2016:trt0f3 0.25 0.58 -0.90 1.40 1351  
## date0f7D18D2016:trt0f3 0.69 0.58 -0.46 1.85 1808  
## date0f7D26D2016:trt0f3 -0.07 0.60 -1.26 1.09 2085  
## date0f7D4D2016:trt0f3 -0.33 0.57 -1.42 0.77 1894  
## date0f8D2D2016:trt0f3 -0.33 0.57 -1.45 0.78 2108  
## date0f6D28D2016:trt0f4 0.12 0.59 -1.04 1.29 2060  
## date0f7D12D2016:trt0f4 -0.49 0.58 -1.61 0.68 1849  
## date0f7D18D2016:trt0f4 -0.53 0.59 -1.66 0.65 1740  
## date0f7D26D2016:trt0f4 -1.17 0.59 -2.34 0.00 2232  
## date0f7D4D2016:trt0f4 -0.54 0.57 -1.62 0.57 2005  
## date0f8D2D2016:trt0f4 -0.35 0.58 -1.48 0.80 2061  
## Rhat  
## Intercept 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016:trt0f2 1.00  
## date0f7D12D2016:trt0f2 1.00  
## date0f7D18D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f7D4D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f6D28D2016:trt0f3 1.00  
## date0f7D12D2016:trt0f3 1.00  
## date0f7D18D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f7D4D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f6D28D2016:trt0f4 1.00  
## date0f7D12D2016:trt0f4 1.00  
## date0f7D18D2016:trt0f4 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f7D4D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 1.56 0.19 1.22 1.95 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(chirobio,type="hist")

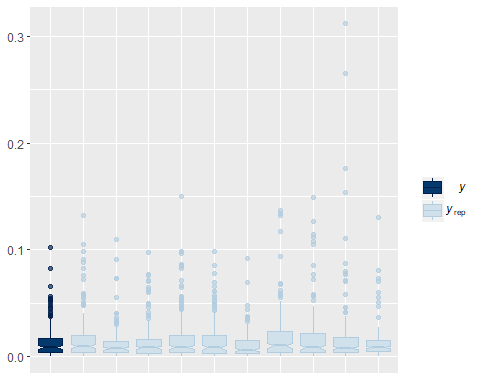
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(chirobio,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Libellulid Biomass

Model

get\_prior(bio\_odo~trt0f+date0f+date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f7D12D2016   
## 4 b date0f7D18D2016   
## 5 b date0f7D26D2016   
## 6 b date0f7D4D2016   
## 7 b date0f8D2D2016   
## 8 b trt0f2   
## 9 b trt0f2:date0f6D28D2016   
## 10 b trt0f2:date0f7D12D2016   
## 11 b trt0f2:date0f7D18D2016   
## 12 b trt0f2:date0f7D26D2016   
## 13 b trt0f2:date0f7D4D2016   
## 14 b trt0f2:date0f8D2D2016   
## 15 b trt0f3   
## 16 b trt0f3:date0f6D28D2016   
## 17 b trt0f3:date0f7D12D2016   
## 18 b trt0f3:date0f7D18D2016   
## 19 b trt0f3:date0f7D26D2016   
## 20 b trt0f3:date0f7D4D2016   
## 21 b trt0f3:date0f8D2D2016   
## 22 b trt0f4   
## 23 b trt0f4:date0f6D28D2016   
## 24 b trt0f4:date0f7D12D2016   
## 25 b trt0f4:date0f7D18D2016   
## 26 b trt0f4:date0f7D26D2016   
## 27 b trt0f4:date0f7D4D2016   
## 28 b trt0f4:date0f8D2D2016   
## 29 student\_t(3, -21, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## 33 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33

odobio<-brm(bio\_odo~trt0f+date0f+date0f\*trt0f +(1|tank),data=biodivnew,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.15 seconds (Warm-up)  
## 5.938 seconds (Sampling)  
## 11.088 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.957 seconds (Warm-up)  
## 4.437 seconds (Sampling)  
## 10.394 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.22 seconds (Warm-up)  
## 5.546 seconds (Sampling)  
## 10.766 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.214 seconds (Warm-up)  
## 5.729 seconds (Sampling)  
## 10.943 seconds (Total)

Model Checks

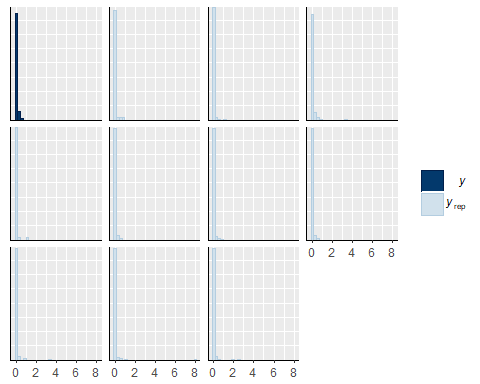
print(odobio)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: bio\_odo ~ trt0f + date0f + date0f \* trt0f + (1 | tank)   
## Data: biodivnew (Number of observations: 164)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.23 0.18 0.01 0.69 3029 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept -14.21 1.00 -16.13 -12.23 1843  
## trt0f2 -0.61 0.91 -2.39 1.17 2481  
## trt0f3 -0.94 0.91 -2.70 0.88 2882  
## trt0f4 -0.70 0.92 -2.50 1.06 2608  
## date0f6D28D2016 -4.54 1.24 -6.94 -2.08 3359  
## date0f7D12D2016 -4.67 1.20 -6.97 -2.23 2787  
## date0f7D18D2016 9.77 1.08 7.66 11.86 2392  
## date0f7D26D2016 11.00 1.06 8.92 13.03 2362  
## date0f7D4D2016 -4.68 1.21 -7.00 -2.29 3011  
## date0f8D2D2016 10.79 1.08 8.72 12.90 2021  
## trt0f2:date0f6D28D2016 -0.58 1.39 -3.23 2.29 4000  
## trt0f3:date0f6D28D2016 -0.46 1.34 -3.10 2.03 4000  
## trt0f4:date0f6D28D2016 -0.60 1.42 -3.33 2.20 4000  
## trt0f2:date0f7D12D2016 -0.59 1.36 -3.13 2.09 4000  
## trt0f3:date0f7D12D2016 -0.36 1.36 -2.96 2.37 4000  
## trt0f4:date0f7D12D2016 -0.52 1.39 -3.15 2.28 3335  
## trt0f2:date0f7D18D2016 2.77 1.16 0.49 5.02 3126  
## trt0f3:date0f7D18D2016 0.90 1.24 -1.53 3.39 4000  
## trt0f4:date0f7D18D2016 2.45 1.18 0.14 4.74 3356  
## trt0f2:date0f7D26D2016 1.33 1.16 -0.92 3.65 4000  
## trt0f3:date0f7D26D2016 2.40 1.16 0.19 4.78 3582  
## trt0f4:date0f7D26D2016 1.68 1.18 -0.54 4.06 3033  
## trt0f2:date0f7D4D2016 -0.58 1.36 -3.13 2.15 4000  
## trt0f3:date0f7D4D2016 -0.35 1.35 -2.91 2.38 3596  
## trt0f4:date0f7D4D2016 -0.51 1.40 -3.13 2.32 3520  
## trt0f2:date0f8D2D2016 1.54 1.16 -0.74 3.92 3367  
## trt0f3:date0f8D2D2016 1.35 1.17 -0.91 3.73 4000  
## trt0f4:date0f8D2D2016 1.18 1.18 -1.07 3.54 3310  
## Rhat  
## Intercept 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2:date0f6D28D2016 1.00  
## trt0f3:date0f6D28D2016 1.00  
## trt0f4:date0f6D28D2016 1.00  
## trt0f2:date0f7D12D2016 1.00  
## trt0f3:date0f7D12D2016 1.00  
## trt0f4:date0f7D12D2016 1.00  
## trt0f2:date0f7D18D2016 1.00  
## trt0f3:date0f7D18D2016 1.00  
## trt0f4:date0f7D18D2016 1.00  
## trt0f2:date0f7D26D2016 1.00  
## trt0f3:date0f7D26D2016 1.00  
## trt0f4:date0f7D26D2016 1.00  
## trt0f2:date0f7D4D2016 1.00  
## trt0f3:date0f7D4D2016 1.00  
## trt0f4:date0f7D4D2016 1.00  
## trt0f2:date0f8D2D2016 1.00  
## trt0f3:date0f8D2D2016 1.00  
## trt0f4:date0f8D2D2016 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 0.19 0.02 0.15 0.22 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(odobio,type="hist")

## Using 10 posterior samples for ppc type 'hist' by default.

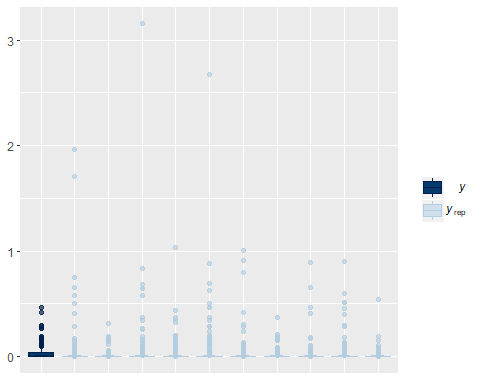
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(odobio,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.

## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.



Example of Comparison among Treatments

Extract posterior samples and add appropriate offsets from the intercept to get estimates of abundance for each treatment on each date

mbiopost<-posterior\_samples(mbiobrm)  
str(mbiopost)

## 'data.frame': 4000 obs. of 55 variables:  
## $ b\_Intercept : num -4.33 -4.47 -4.65 -4.69 -4.48 ...  
## $ b\_date0f6D28D2016 : num 0.367 1.242 0.881 0.926 1.173 ...  
## $ b\_date0f7D12D2016 : num 0.0765 -0.2223 -0.2335 -0.0595 0.5416 ...  
## $ b\_date0f7D18D2016 : num 2.31 1.79 2.57 2.42 2.42 ...  
## $ b\_date0f7D26D2016 : num 1.51 2.67 1.76 1.71 1.46 ...  
## $ b\_date0f7D4D2016 : num 1.41 1.68 1.67 1.72 2.69 ...  
## $ b\_date0f8D2D2016 : num 2.38 1.48 1.83 1.96 1.72 ...  
## $ b\_trt0f2 : num -0.1295 -0.6173 -0.2268 0.1017 -0.0202 ...  
## $ b\_trt0f3 : num -0.3894 -0.2065 -0.2968 -0.3996 0.0531 ...  
## $ b\_trt0f4 : num 0.1302 0.0768 -0.1391 0.0435 -0.5263 ...  
## $ b\_date0f6D28D2016:trt0f2: num 0.461 -0.153 0.813 0.529 -0.699 ...  
## $ b\_date0f7D12D2016:trt0f2: num -0.211 0.493 0.936 0.568 -0.621 ...  
## $ b\_date0f7D18D2016:trt0f2: num 0.3721 1.3834 -0.0511 -0.3759 -0.1888 ...  
## $ b\_date0f7D26D2016:trt0f2: num 0.41291 0.00503 0.78865 0.39233 0.2567 ...  
## $ b\_date0f7D4D2016:trt0f2 : num -0.2761 0.0943 -0.604 -0.7788 -2.1241 ...  
## $ b\_date0f8D2D2016:trt0f2 : num 0.29 1.467 0.403 0.641 0.5 ...  
## $ b\_date0f6D28D2016:trt0f3: num 0.9203 -0.4594 -0.0771 0.2204 0.2712 ...  
## $ b\_date0f7D12D2016:trt0f3: num 0.648 0.843 0.82 1.322 -0.153 ...  
## $ b\_date0f7D18D2016:trt0f3: num -1.593 -1.499 -0.899 -1.337 -2.634 ...  
## $ b\_date0f7D26D2016:trt0f3: num 1.352 0.768 1.75 1.426 1.349 ...  
## $ b\_date0f7D4D2016:trt0f3 : num 0.663 -0.107 0.347 0.292 -0.993 ...  
## $ b\_date0f8D2D2016:trt0f3 : num -0.4089 0.4552 0.2038 0.9297 -0.0214 ...  
## $ b\_date0f6D28D2016:trt0f4: num 0.3874 -0.5743 0.237 0.5329 0.0559 ...  
## $ b\_date0f7D12D2016:trt0f4: num -0.722 -0.739 0.331 0.687 -0.374 ...  
## $ b\_date0f7D18D2016:trt0f4: num 0.7656 0.4274 -0.6954 -0.7828 0.0976 ...  
## $ b\_date0f7D26D2016:trt0f4: num 0.616 -0.225 1.069 0.858 1.192 ...  
## $ b\_date0f7D4D2016:trt0f4 : num -1.0942 -1.2423 -0.0864 -0.3957 -1.2521 ...  
## $ b\_date0f8D2D2016:trt0f4 : num -0.378 0.245 0.608 0.518 1.47 ...  
## $ sd\_tank\_\_Intercept : num 0.334 0.308 0.251 0.219 0.165 ...  
## $ shape : num 0.923 0.884 0.937 0.964 0.834 ...  
## $ r\_tank[1,Intercept] : num 0.2551 0.4826 0.1461 0.2232 0.0588 ...  
## $ r\_tank[2,Intercept] : num -0.0707 0.4713 -0.0544 -0.1764 -0.1402 ...  
## $ r\_tank[3,Intercept] : num -0.21 -0.371 0.396 0.326 -0.335 ...  
## $ r\_tank[4,Intercept] : num -0.4015 -0.2146 0.01 0.0208 -0.2517 ...  
## $ r\_tank[5,Intercept] : num -0.6226 0.1963 -0.0507 0.1167 0.1078 ...  
## $ r\_tank[6,Intercept] : num 0.109 -0.102 -0.129 -0.181 -0.242 ...  
## $ r\_tank[7,Intercept] : num 0.1743 0.0722 0.105 0.264 -0.1598 ...  
## $ r\_tank[8,Intercept] : num -0.4181 -0.0701 0.1053 0.1229 -0.1475 ...  
## $ r\_tank[9,Intercept] : num 0.0248 -0.3453 0.3671 0.3349 -0.0487 ...  
## $ r\_tank[10,Intercept] : num -0.4393 -0.4583 0.1661 0.0155 -0.2383 ...  
## $ r\_tank[11,Intercept] : num 0.4191 -0.1479 0.3675 0.3796 -0.0893 ...  
## $ r\_tank[12,Intercept] : num 0.311 -0.039 0.526 0.302 0.331 ...  
## $ r\_tank[13,Intercept] : num 0.0583 -0.4111 0.6958 0.4499 -0.1504 ...  
## $ r\_tank[14,Intercept] : num -0.0491 0.2778 0.2197 0.0662 -0.0301 ...  
## $ r\_tank[15,Intercept] : num 0.341 0.659 0.175 0.103 0.17 ...  
## $ r\_tank[16,Intercept] : num -0.452 0.101 -0.197 -0.203 0.193 ...  
## $ r\_tank[17,Intercept] : num -0.36985 -0.00465 -0.33889 -0.2933 -0.24675 ...  
## $ r\_tank[18,Intercept] : num -0.2524 0.0893 -0.2314 -0.1493 0.178 ...  
## $ r\_tank[19,Intercept] : num 0.2441 -0.0411 0.0847 0.1981 0.1061 ...  
## $ r\_tank[20,Intercept] : num 0.2474 0.5943 -0.0729 0.0896 0.0688 ...  
## $ r\_tank[21,Intercept] : num -0.3703 -0.1194 -0.0959 0.0805 0.1203 ...  
## $ r\_tank[22,Intercept] : num 0.0282 0.1959 0.083 0.1215 0.2098 ...  
## $ r\_tank[23,Intercept] : num 0.0496 -0.3462 0.1984 0.2029 -0.2108 ...  
## $ r\_tank[24,Intercept] : num -0.558 -0.453 -0.172 -0.277 -0.205 ...  
## $ lp\_\_ : num 292 291 288 286 281 ...

btrt1\_3<-mbiopost$b\_Intercept+mbiopost$b\_date0f7D4D2016 #posterior treatment 1 on date 3  
btrt2\_3<-mbiopost$b\_Intercept+mbiopost$b\_date0f7D4D2016+mbiopost$b\_trt0f2+mbiopost$`b\_date0f7D4D2016:trt0f2`  
  
#Trt 1 & 2 on date 3#  
btrt1\_2\_d3<-(((exp(btrt1\_3)\*2.78)/4))-(((exp(btrt2\_3)\*2.78)/4))  
mean(btrt1\_2\_d3)

## [1] 0.01980513

quantile(btrt1\_2\_d3,probs=c(0.025,0.975))

## 2.5% 97.5%   
## -0.02171626 0.07281696

sum(btrt1\_2\_d3>0)/4000 #87.8%

## [1] 0.857

Preparing to Plot

Create data frame for total, chironomid, and libellulid biomass, edit date as a properly ordered factor so that it plots dates in the right order, convert estimates and upper and lower quantiles to g/m2/d.

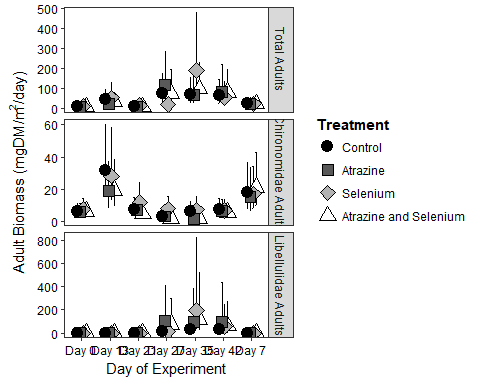
#TOTAL  
totbplot<-marginal\_effects(mbiobrm, robust=FALSE)  
totbplot<-as.data.frame(totbplot$`date0f:trt0f`)  
totbplot$date3<-factor(totbplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
totbplot$est2<-(totbplot$estimate\_\_)\*2.78/4  
totbplot$upper2<-(totbplot$upper\_\_)\*2.78/4  
totbplot$lower2<-(totbplot$lower\_\_)\*2.78/4  
  
#CHIRONOMID  
chirbplot<-marginal\_effects(chirobio, robust=FALSE)  
chirbplot<-as.data.frame(chirbplot$`date0f:trt0f`)  
chirbplot$date2<-factor(chirbplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
chirbplot$est2<-(chirbplot$estimate\_\_)\*2.78/4  
chirbplot$upper2<-(chirbplot$upper\_\_)\*2.78/4  
chirbplot$lower2<-(chirbplot$lower\_\_)\*2.78/4  
  
#LIBELLULID  
odobplot<-marginal\_effects(odobio, effects="date0f:trt0f", robust=FALSE)  
odobplot<-as.data.frame(odobplot$`date0f:trt0f`)  
odobplot$date2<-factor(odobplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
odobplot$est2<-(odobplot$estimate\_\_)\*2.78/4  
odobplot$upper2<-(odobplot$upper\_\_)\*2.78/4  
odobplot$lower2<-(odobplot$lower\_\_)\*2.78/4

Combine all three data frames to one and convert data to mgDM, allowing for creation of a facetted plot, change date to day of experiment

chirbplot$group<-"Chironomidae Adults"  
odobplot$group<-"Libellulidae Adults"  
totbplot$group<-"Total Adults"  
  
totbplot$date2<-totbplot$date3  
totbplot$date3<-NULL  
totbplot$totbio<-NULL  
chirbplot$biochiro<-NULL  
odobplot$bio\_odo<-NULL  
adubiomass<-rbind(totbplot, chirbplot, odobplot)  
  
adubiomass$group2<-factor(adubiomass$group, levels= c("Total Adults", "Chironomidae Adults", "Libellulidae Adults"))  
adubiomass$est3<-(adubiomass$est2)\*1000  
adubiomass$lower3<-(adubiomass$lower2)\*1000  
adubiomass$upper3<-(adubiomass$upper2)\*1000  
  
adubiomass$date3<-ifelse(adubiomass$date2=="6/20/2016", "Day 0",  
 ifelse(adubiomass$date2=="6/28/2016", "Day 7",  
 ifelse(adubiomass$date2=="7/4/2016", "Day 13",  
 ifelse(adubiomass$date2=="7/12/2016", "Day 21",  
 ifelse(adubiomass$date2=="7/18/2016", "Day 27",  
 ifelse(adubiomass$date2=="7/26/2016", "Day 35", "Day 42"))))))

Create plot

ggplot(adubiomass,aes(x=date3,y=est3, fill=trt0f, shape=trt0f))+  
 geom\_errorbar(aes(ymin=lower3, ymax=upper3), width=.1, position=position\_dodge(width = 0.4))+  
 geom\_point(position=position\_dodge(width = 0.4), size=4)+  
 facet\_grid(group2~., scales="free")+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Adult Biomass (mgDM/m"^2,"/day)")))+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_bw()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"),  
 panel.grid = element\_blank())



Adult Weekly Abundance

Adult abundance data also comes from the biodivnew data file, which is already imported above. Data are in # insects/0.36m2/4days.

Model

get\_prior(totno~date0f+trt0f+date0f\*trt0f +(1|tank),data=biodivnew,family=poisson(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f6D28D2016:trt0f2   
## 4 b date0f6D28D2016:trt0f3   
## 5 b date0f6D28D2016:trt0f4   
## 6 b date0f7D12D2016   
## 7 b date0f7D12D2016:trt0f2   
## 8 b date0f7D12D2016:trt0f3   
## 9 b date0f7D12D2016:trt0f4   
## 10 b date0f7D18D2016   
## 11 b date0f7D18D2016:trt0f2   
## 12 b date0f7D18D2016:trt0f3   
## 13 b date0f7D18D2016:trt0f4   
## 14 b date0f7D26D2016   
## 15 b date0f7D26D2016:trt0f2   
## 16 b date0f7D26D2016:trt0f3   
## 17 b date0f7D26D2016:trt0f4   
## 18 b date0f7D4D2016   
## 19 b date0f7D4D2016:trt0f2   
## 20 b date0f7D4D2016:trt0f3   
## 21 b date0f7D4D2016:trt0f4   
## 22 b date0f8D2D2016   
## 23 b date0f8D2D2016:trt0f2   
## 24 b date0f8D2D2016:trt0f3   
## 25 b date0f8D2D2016:trt0f4   
## 26 b trt0f2   
## 27 b trt0f3   
## 28 b trt0f4   
## 29 student\_t(3, 3, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32

abundadult<-brm(totno~date0f+trt0f+date0f\*trt0f +(1|tank),data=biodivnew,family=poisson(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,5),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 25.727 seconds (Warm-up)  
## 28.442 seconds (Sampling)  
## 54.169 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 23.805 seconds (Warm-up)  
## 26.787 seconds (Sampling)  
## 50.592 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 33.043 seconds (Warm-up)  
## 24.403 seconds (Sampling)  
## 57.446 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 23.51 seconds (Warm-up)  
## 29.202 seconds (Sampling)  
## 52.712 seconds (Total)

Model Checks

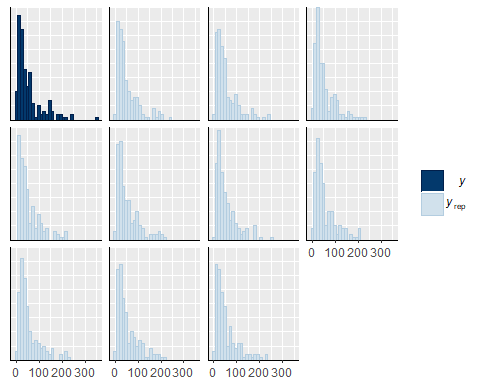
print(abundadult)

## Family: poisson   
## Links: mu = log   
## Formula: totno ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: biodivnew (Number of observations: 164)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.46 0.08 0.33 0.64 1776 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept 3.14 0.20 2.73 3.54 1949  
## date0f6D28D2016 1.33 0.09 1.16 1.52 1726  
## date0f7D12D2016 0.47 0.11 0.27 0.69 2032  
## date0f7D18D2016 -0.18 0.12 -0.42 0.05 2193  
## date0f7D26D2016 0.17 0.11 -0.05 0.39 2107  
## date0f7D4D2016 1.91 0.09 1.75 2.09 1696  
## date0f8D2D2016 1.06 0.10 0.88 1.26 1703  
## trt0f2 0.41 0.29 -0.15 0.99 2414  
## trt0f3 0.14 0.28 -0.40 0.72 2268  
## trt0f4 0.06 0.28 -0.52 0.61 1864  
## date0f6D28D2016:trt0f2 -0.61 0.13 -0.86 -0.37 2179  
## date0f7D12D2016:trt0f2 -0.13 0.14 -0.41 0.14 2384  
## date0f7D18D2016:trt0f2 -0.36 0.16 -0.69 -0.05 2806  
## date0f7D26D2016:trt0f2 -1.19 0.17 -1.53 -0.85 2927  
## date0f7D4D2016:trt0f2 -1.02 0.12 -1.26 -0.79 2127  
## date0f8D2D2016:trt0f2 -0.86 0.13 -1.13 -0.61 2197  
## date0f6D28D2016:trt0f3 0.12 0.12 -0.12 0.37 2298  
## date0f7D12D2016:trt0f3 -0.55 0.15 -0.84 -0.25 2724  
## date0f7D18D2016:trt0f3 0.09 0.16 -0.22 0.41 3034  
## date0f7D26D2016:trt0f3 -0.10 0.16 -0.40 0.21 2743  
## date0f7D4D2016:trt0f3 -0.62 0.12 -0.85 -0.38 2292  
## date0f8D2D2016:trt0f3 -0.43 0.13 -0.69 -0.18 2426  
## date0f6D28D2016:trt0f4 -0.08 0.13 -0.33 0.17 1930  
## date0f7D12D2016:trt0f4 -0.61 0.16 -0.92 -0.31 2565  
## date0f7D18D2016:trt0f4 -0.63 0.19 -1.00 -0.26 2808  
## date0f7D26D2016:trt0f4 -0.82 0.17 -1.16 -0.49 2564  
## date0f7D4D2016:trt0f4 -0.50 0.12 -0.74 -0.26 1999  
## date0f8D2D2016:trt0f4 -0.75 0.14 -1.03 -0.48 2129  
## Rhat  
## Intercept 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016:trt0f2 1.00  
## date0f7D12D2016:trt0f2 1.00  
## date0f7D18D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f7D4D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f6D28D2016:trt0f3 1.00  
## date0f7D12D2016:trt0f3 1.00  
## date0f7D18D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f7D4D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f6D28D2016:trt0f4 1.00  
## date0f7D12D2016:trt0f4 1.00  
## date0f7D18D2016:trt0f4 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f7D4D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(abundadult,type="hist")

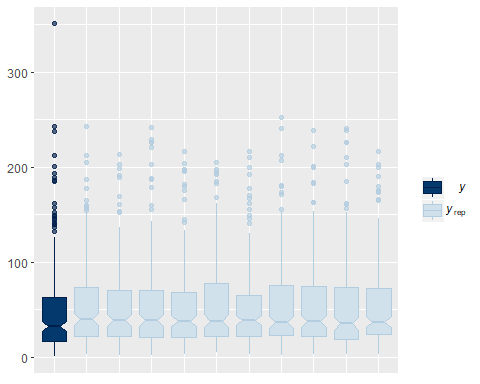
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(abundadult,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Comparisons on a Given Date

Extract posterior samples and add appropriate offsets from the intercept to get estimates of abundance for each treatment on each date

m2post<-posterior\_samples(abundadult)  
str(m2post)

## 'data.frame': 4000 obs. of 54 variables:  
## $ b\_Intercept : num 3.1 3.03 3.18 2.87 3.19 ...  
## $ b\_date0f6D28D2016 : num 1.39 1.29 1.27 1.41 1.38 ...  
## $ b\_date0f7D12D2016 : num 0.55 0.524 0.359 0.637 0.507 ...  
## $ b\_date0f7D18D2016 : num -0.2305 -0.1881 -0.3522 0.0517 -0.1784 ...  
## $ b\_date0f7D26D2016 : num 0.252 0.195 0.192 0.194 0.482 ...  
## $ b\_date0f7D4D2016 : num 1.9 1.9 1.85 2.02 1.95 ...  
## $ b\_date0f8D2D2016 : num 1.11 1.11 1.01 1.18 1.11 ...  
## $ b\_trt0f2 : num 0.518 0.614 0.642 0.546 0.254 ...  
## $ b\_trt0f3 : num 0.106 0.251 0.187 0.656 0.619 ...  
## $ b\_trt0f4 : num 0.033 0.123 0.156 0.589 -0.065 ...  
## $ b\_date0f6D28D2016:trt0f2: num -0.587 -0.568 -0.726 -0.693 -0.76 ...  
## $ b\_date0f7D12D2016:trt0f2: num -0.147 -0.181 -0.137 -0.191 -0.167 ...  
## $ b\_date0f7D18D2016:trt0f2: num -0.371 -0.476 -0.361 -0.379 -0.362 ...  
## $ b\_date0f7D26D2016:trt0f2: num -1.36 -1.24 -1.39 -1.03 -1.36 ...  
## $ b\_date0f7D4D2016:trt0f2 : num -1.06 -1.01 -1.04 -1.08 -1.11 ...  
## $ b\_date0f8D2D2016:trt0f2 : num -0.883 -0.958 -0.893 -0.877 -0.942 ...  
## $ b\_date0f6D28D2016:trt0f3: num 0.2374 0.3141 0.2157 -0.0611 -0.0723 ...  
## $ b\_date0f7D12D2016:trt0f3: num -0.466 -0.536 -0.319 -0.747 -0.747 ...  
## $ b\_date0f7D18D2016:trt0f3: num 0.341 0.167 0.234 -0.179 -0.179 ...  
## $ b\_date0f7D26D2016:trt0f3: num -0.00453 -0.01965 -0.14115 -0.12741 -0.43004 ...  
## $ b\_date0f7D4D2016:trt0f3 : num -0.512 -0.551 -0.586 -0.816 -0.742 ...  
## $ b\_date0f8D2D2016:trt0f3 : num -0.447 -0.293 -0.211 -0.641 -0.496 ...  
## $ b\_date0f6D28D2016:trt0f4: num -0.2337 0.049 0.0124 -0.1144 -0.1413 ...  
## $ b\_date0f7D12D2016:trt0f4: num -0.662 -0.786 -0.517 -0.801 -0.884 ...  
## $ b\_date0f7D18D2016:trt0f4: num -0.617 -0.438 -0.728 -0.967 -0.588 ...  
## $ b\_date0f7D26D2016:trt0f4: num -1.023 -0.839 -0.85 -0.632 -1.17 ...  
## $ b\_date0f7D4D2016:trt0f4 : num -0.594 -0.523 -0.407 -0.578 -0.569 ...  
## $ b\_date0f8D2D2016:trt0f4 : num -0.749 -0.944 -0.759 -0.933 -0.835 ...  
## $ sd\_tank\_\_Intercept : num 0.385 0.412 0.52 0.425 0.515 ...  
## $ r\_tank[1,Intercept] : num 0.207 0.2172 0.2362 0.3602 0.0587 ...  
## $ r\_tank[2,Intercept] : num 0.0285 -0.0946 -0.2134 0.2073 0.1888 ...  
## $ r\_tank[3,Intercept] : num -0.145 -0.139 -0.147 -0.289 -0.35 ...  
## $ r\_tank[4,Intercept] : num -0.0748 0.0934 -0.0474 0.1431 -0.198 ...  
## $ r\_tank[5,Intercept] : num 0.476 0.484 0.424 0.325 0.1 ...  
## $ r\_tank[6,Intercept] : num -0.384 -0.193 -0.361 -0.142 -0.38 ...  
## $ r\_tank[7,Intercept] : num 0.00464 -0.04218 0.06727 0.17851 0.29936 ...  
## $ r\_tank[8,Intercept] : num -0.134 -0.16 -0.137 -0.199 -0.536 ...  
## $ r\_tank[9,Intercept] : num 0.1488 0.1792 0.0573 -0.1129 0.2073 ...  
## $ r\_tank[10,Intercept] : num 0.3178 0.1984 0.0686 -0.0398 0.2491 ...  
## $ r\_tank[11,Intercept] : num 0.275 0.375 0.358 0.551 0.153 ...  
## $ r\_tank[12,Intercept] : num 0.4076 0.2994 0.2749 0.2844 0.0854 ...  
## $ r\_tank[13,Intercept] : num -0.747 -0.533 -0.595 -0.539 -0.498 ...  
## $ r\_tank[14,Intercept] : num -0.38 -0.626 -0.7 -0.874 -0.353 ...  
## $ r\_tank[15,Intercept] : num 0.924 0.8 0.63 0.584 0.936 ...  
## $ r\_tank[16,Intercept] : num 0.609 0.58 0.506 0.742 0.813 ...  
## $ r\_tank[17,Intercept] : num 0.2129 0.3019 0.2037 0.3441 0.0395 ...  
## $ r\_tank[18,Intercept] : num -0.16113 -0.0945 -0.17269 0.00864 0.22992 ...  
## $ r\_tank[19,Intercept] : num -0.332 -0.193 -0.214 -0.195 -0.442 ...  
## $ r\_tank[20,Intercept] : num -0.0891 -0.2385 -0.2268 -0.2003 -0.5485 ...  
## $ r\_tank[21,Intercept] : num -0.321 -0.4 -0.477 -0.104 -0.144 ...  
## $ r\_tank[22,Intercept] : num 0.201 0.0796 -0.0116 -0.2617 0.3053 ...  
## $ r\_tank[23,Intercept] : num -0.85 -0.83 -0.831 -0.989 -1.153 ...  
## $ r\_tank[24,Intercept] : num -0.654 -0.698 -0.756 -1.042 -0.509 ...  
## $ lp\_\_ : num -1718 -1715 -1710 -1714 -1721 ...

#estimate posteriors for each date\*trt combination.   
  
trt1\_2<-m2post$b\_Intercept+m2post$b\_date0f2 #posterior treatment 1 on date 2  
trt1\_3<-m2post$b\_Intercept+m2post$b\_date0f3 #posterior treatment 1 on date 3  
trt1\_4<-m2post$b\_Intercept+m2post$b\_date0f4 #posterior treatment 1 on date 3  
trt1\_5<-m2post$b\_Intercept+m2post$b\_date0f5 #posterior treatment 1 on date 3  
trt1\_6<-m2post$b\_Intercept+m2post$b\_date0f6 #posterior treatment 1 on date 3  
trt1\_7<-m2post$b\_Intercept+m2post$b\_date0f7 #posterior treatment 1 on date 3  
  
trt2\_2<-m2post$b\_Intercept+m2post$b\_date0f2+m2post$b\_trt0f2+m2post$`b\_date0f2:trt0f2`  
trt2\_3<-m2post$b\_Intercept+m2post$b\_date0f3+m2post$b\_trt0f2+m2post$`b\_date0f3:trt0f2`  
trt2\_4<-m2post$b\_Intercept+m2post$b\_date0f4+m2post$b\_trt0f2+m2post$`b\_date0f4:trt0f2`  
trt2\_5<-m2post$b\_Intercept+m2post$b\_date0f5+m2post$b\_trt0f2+m2post$`b\_date0f5:trt0f2`  
trt2\_6<-m2post$b\_Intercept+m2post$b\_date0f6+m2post$b\_trt0f2+m2post$`b\_date0f6:trt0f2`  
trt2\_7<-m2post$b\_Intercept+m2post$b\_date0f7+m2post$b\_trt0f2+m2post$`b\_date0f7:trt0f2`  
  
trt3\_2<-m2post$b\_Intercept+m2post$b\_date0f2+m2post$b\_trt0f3+m2post$`b\_date0f2:trt0f3`  
trt3\_3<-m2post$b\_Intercept+m2post$b\_date0f3+m2post$b\_trt0f3+m2post$`b\_date0f3:trt0f3`  
trt3\_4<-m2post$b\_Intercept+m2post$b\_date0f4+m2post$b\_trt0f3+m2post$`b\_date0f4:trt0f3`  
trt3\_5<-m2post$b\_Intercept+m2post$b\_date0f5+m2post$b\_trt0f3+m2post$`b\_date0f5:trt0f3`  
trt3\_6<-m2post$b\_Intercept+m2post$b\_date0f6+m2post$b\_trt0f3+m2post$`b\_date0f6:trt0f3`  
trt3\_7<-m2post$b\_Intercept+m2post$b\_date0f7+m2post$b\_trt0f3+m2post$`b\_date0f7:trt0f3`  
  
trt4\_2<-m2post$b\_Intercept+m2post$b\_date0f2+m2post$b\_trt0f4+m2post$`b\_date0f2:trt0f4`   
trt4\_3<-m2post$b\_Intercept+m2post$b\_date0f3+m2post$b\_trt0f4+m2post$`b\_date0f3:trt0f4`  
trt4\_4<-m2post$b\_Intercept+m2post$b\_date0f4+m2post$b\_trt0f4+m2post$`b\_date0f4:trt0f4`   
trt4\_5<-m2post$b\_Intercept+m2post$b\_date0f5+m2post$b\_trt0f4+m2post$`b\_date0f5:trt0f4`   
trt4\_6<-m2post$b\_Intercept+m2post$b\_date0f6+m2post$b\_trt0f4+m2post$`b\_date0f6:trt0f4`   
trt4\_7<-m2post$b\_Intercept+m2post$b\_date0f7+m2post$b\_trt0f4+m2post$`b\_date0f7:trt0f4`

Comparison example

#Trt 1 & 4   
trt1\_4\_d3<-(((exp(trt1\_3)\*2.78)/4))-(((exp(trt4\_3)\*2.78)/4))   
mean(trt1\_4\_d3) #Average of 39 fewer insects emerging per m2 per day relative to the control on date 3

## [1] NaN

quantile(trt1\_4\_d3,probs=c(0.025,0.975))

## 2.5% 97.5%   
## NA NA

sum(trt1\_4\_d3>0)/4000 #We are 95% certain that the difference in emergence between trts 1 and 4 on date 3 was greater than zero.

## [1] 0

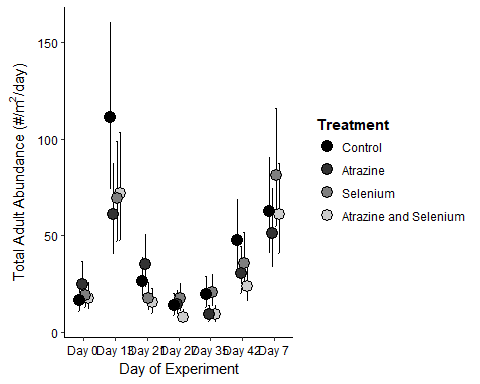
Preparing to plot

Create data frame, edit date as a properly ordered factor so that it plots dates in the right order, change date to day of experiment, convert estimates and upper and lower quantiles to #/m2/day.

aaplot<-marginal\_effects(abundadult, robust=FALSE)  
aaplot<-as.data.frame(aaplot$`date0f:trt0f`)  
aaplot$date2<-factor(aaplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
aaplot$est2<-(aaplot$estimate\_\_)\*2.78/4  
aaplot$upper2<-(aaplot$upper\_\_)\*2.78/4  
aaplot$lower2<-(aaplot$lower\_\_)\*2.78/4  
  
aaplot$date3<-ifelse(aaplot$date2=="6/20/2016", "Day 0",  
 ifelse(aaplot$date2=="6/28/2016", "Day 7",  
 ifelse(aaplot$date2=="7/4/2016", "Day 13",  
 ifelse(aaplot$date2=="7/12/2016", "Day 21",  
 ifelse(aaplot$date2=="7/18/2016", "Day 27",  
 ifelse(aaplot$date2=="7/26/2016", "Day 35", "Day 42"))))))

Plot

ggplot(aaplot,aes(x=date3,y=est2, fill=trt0f))+  
 geom\_errorbar(aes(ymin=lower2, ymax=upper2), width=.1, position=position\_dodge(width=0.4))+  
 geom\_point(position=position\_dodge(width=0.4), size=4, shape=21)+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Total Adult Abundance (#/m"^2,"/day)")))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey20","grey50","grey80"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))



Adult Cumulative Abundance

From the above model, we use the posterior distribution to estimate the cumulative emergence from each treatment and make comparisons.

trt1\_tot<-(((exp(trt1\_2)\*2.78)/4))+(((exp(trt1\_3)\*2.78)/4))+(((exp(trt1\_4)\*2.78)/4))+(((exp(trt1\_5)\*2.78)/4))+(((exp(trt1\_6)\*2.78)/4))+(((exp(trt1\_7)\*2.78)/4))  
trt2\_tot<-(((exp(trt2\_2)\*2.78)/4))+(((exp(trt2\_3)\*2.78)/4))+(((exp(trt2\_4)\*2.78)/4))+(((exp(trt2\_5)\*2.78)/4))+(((exp(trt2\_6)\*2.78)/4))+(((exp(trt2\_7)\*2.78)/4))  
trt3\_tot<-(((exp(trt3\_2)\*2.78)/4))+(((exp(trt3\_3)\*2.78)/4))+(((exp(trt3\_4)\*2.78)/4))+(((exp(trt3\_5)\*2.78)/4))+(((exp(trt3\_6)\*2.78)/4))+(((exp(trt3\_7)\*2.78)/4))  
trt4\_tot<-(((exp(trt4\_2)\*2.78)/4))+(((exp(trt4\_3)\*2.78)/4))+(((exp(trt4\_4)\*2.78)/4))+(((exp(trt4\_5)\*2.78)/4))+(((exp(trt4\_6)\*2.78)/4))+(((exp(trt4\_7)\*2.78)/4))  
  
mean(trt1\_tot) #mean control

## [1] NaN

mean(trt2\_tot) #mean atrazine

## [1] NaN

mean(trt3\_tot) #mean selenium

## [1] NaN

mean(trt4\_tot) #mean combined

## [1] NaN

quantile(trt1\_tot,probs=c(0.025,0.975)) #quantile control

## 2.5% 97.5%   
## NA NA

quantile(trt2\_tot,probs=c(0.025,0.975)) #quantile atrazine

## 2.5% 97.5%   
## NA NA

quantile(trt3\_tot,probs=c(0.025,0.975)) #quantile selenium

## 2.5% 97.5%   
## NA NA

quantile(trt4\_tot,probs=c(0.025,0.975)) #quantile combined

## 2.5% 97.5%   
## NA NA

Example of comparisons among treatments

#fold difference in cumulative emergence  
diff1\_4\_fold<-trt1\_tot/trt4\_tot  
quantile(diff1\_4\_fold, probs=c(0.025,0.975))

## 2.5% 97.5%   
## NA NA

#0.9-2.5 fold change  
  
#Comparing Treatments 1 & 4  
diff1\_4<-trt1\_tot-trt4\_tot #difference between treatments estimated at each of the 4000 iterations of the MCMC.   
mean(diff1\_4) #Average of 90 more insects emerging from trt1 than trt4 per m2 over the 6 week period

## [1] NaN

quantile(diff1\_4,probs=c(0.025,0.975)) #We are 95% certain that the true difference in cumulative emergence between trt1 and 4 was between -28 and 225

## 2.5% 97.5%   
## NA NA

sum(diff1\_4>0)/4000 #There is a 93% probability that the difference in these treatments was > 0

## [1] 0

Larval Biomass

Larval biomass data is from the larvbio2 file and is in mgDM/0.032m2.

urlfile2<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/larvbio2.csv"  
larvbio2<-read.csv(urlfile2)  
larvbio2$date0f<-as.factor(larvbio2$date)  
larvbio2$trt0f<-as.factor(larvbio2$trt)

Total Biomass

Model

get\_prior(totbio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D6D2016   
## 3 b date0f7D6D2016:trt0f2   
## 4 b date0f7D6D2016:trt0f3   
## 5 b date0f7D6D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, 4, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## 17 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17

lbio<-brm(totbio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.057 seconds (Warm-up)  
## 1.683 seconds (Sampling)  
## 3.74 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.364 seconds (Warm-up)  
## 2.241 seconds (Sampling)  
## 4.605 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.468 seconds (Warm-up)  
## 2.409 seconds (Sampling)  
## 4.877 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.162 seconds (Warm-up)  
## 2.267 seconds (Sampling)  
## 4.429 seconds (Total)

Model Checks

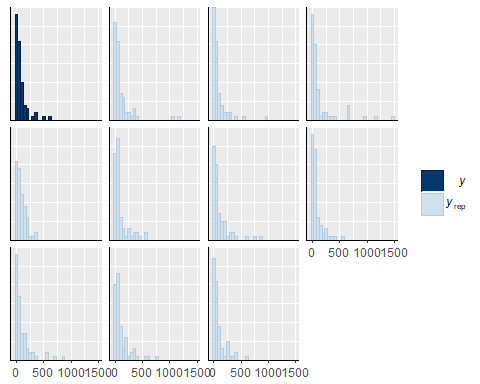
print(lbio)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: totbio ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: larvbio2 (Number of observations: 71)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.52 0.25 0.05 1.01 923 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## Intercept 3.88 0.42 3.09 4.74 1979 1.00  
## date0f7D6D2016 0.78 0.53 -0.23 1.88 2112 1.00  
## date0f8D2D2016 0.70 0.51 -0.28 1.74 2449 1.00  
## trt0f2 -0.27 0.60 -1.46 0.90 2206 1.00  
## trt0f3 -0.06 0.59 -1.19 1.13 2012 1.00  
## trt0f4 -0.18 0.61 -1.32 1.05 2334 1.00  
## date0f7D6D2016:trt0f2 -0.18 0.76 -1.70 1.28 2382 1.00  
## date0f8D2D2016:trt0f2 -0.18 0.75 -1.66 1.26 2543 1.00  
## date0f7D6D2016:trt0f3 -0.59 0.74 -2.06 0.82 2553 1.00  
## date0f8D2D2016:trt0f3 -0.46 0.72 -1.83 0.97 2905 1.00  
## date0f7D6D2016:trt0f4 0.95 0.77 -0.61 2.46 2540 1.00  
## date0f8D2D2016:trt0f4 0.29 0.73 -1.19 1.72 2773 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 1.06 0.20 0.72 1.50 1762 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(lbio,type="hist")

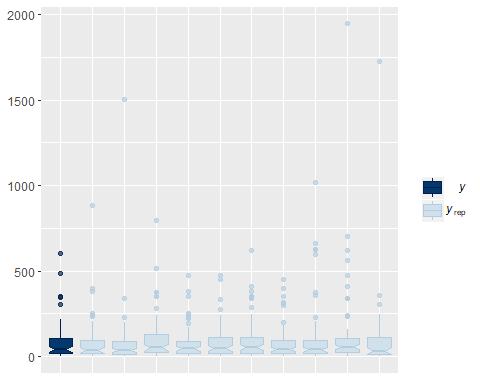
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(lbio,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Chironomid Biomass

Model

get\_prior(chirobio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D6D2016   
## 3 b date0f7D6D2016:trt0f2   
## 4 b date0f7D6D2016:trt0f3   
## 5 b date0f7D6D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, 3, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## 17 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17

chirlbio<-brm(chirobio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.863 seconds (Warm-up)  
## 1.369 seconds (Sampling)  
## 3.232 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.801 seconds (Warm-up)  
## 2.21 seconds (Sampling)  
## 4.011 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.753 seconds (Warm-up)  
## 1.307 seconds (Sampling)  
## 3.06 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.687 seconds (Warm-up)  
## 2.384 seconds (Sampling)  
## 4.071 seconds (Total)

## Warning: There were 1 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help. See  
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

Model Checks

print(chirlbio)

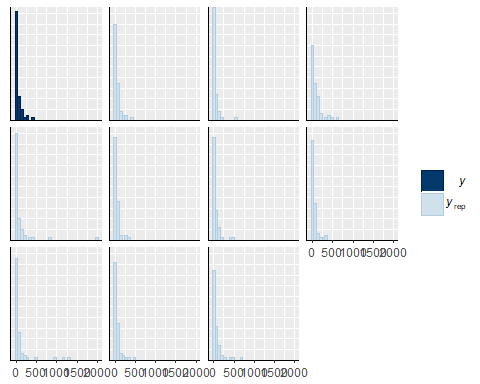
## Warning: There were 1 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help.  
## See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: chirobio ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: larvbio2 (Number of observations: 71)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.33 0.23 0.02 0.85 1071 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## Intercept 3.97 0.45 3.15 4.91 1960 1.00  
## date0f7D6D2016 0.38 0.61 -0.81 1.59 2295 1.00  
## date0f8D2D2016 -0.82 0.60 -2.01 0.39 2472 1.00  
## trt0f2 -0.24 0.65 -1.48 1.05 1987 1.00  
## trt0f3 -0.15 0.63 -1.36 1.11 2629 1.00  
## trt0f4 -0.29 0.64 -1.51 1.03 2255 1.00  
## date0f7D6D2016:trt0f2 0.01 0.89 -1.75 1.74 2425 1.00  
## date0f8D2D2016:trt0f2 -0.49 0.86 -2.14 1.22 2877 1.00  
## date0f7D6D2016:trt0f3 -0.48 0.87 -2.22 1.29 2188 1.00  
## date0f8D2D2016:trt0f3 -0.26 0.86 -1.98 1.42 2838 1.00  
## date0f7D6D2016:trt0f4 1.01 0.89 -0.69 2.73 2749 1.00  
## date0f8D2D2016:trt0f4 -1.28 0.87 -3.00 0.43 2644 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 0.66 0.10 0.48 0.88 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(chirlbio,type="hist")

## Using 10 posterior samples for ppc type 'hist' by default.

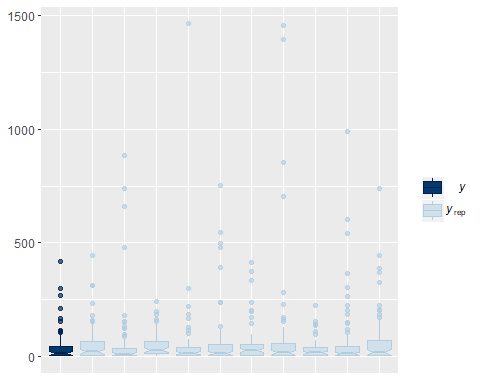
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(chirlbio,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.

## notch went outside hinges. Try setting notch=FALSE.



Libellulid Biomass

Model

get\_prior(odobio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D6D2016   
## 3 b date0f7D6D2016:trt0f2   
## 4 b date0f7D6D2016:trt0f3   
## 5 b date0f7D6D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, 2, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## 17 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17

odolbio<-brm(odobio~date0f+trt0f+date0f\*trt0f +(1|tank),data=larvbio2,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,2),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.916 seconds (Warm-up)  
## 1.262 seconds (Sampling)  
## 3.178 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.783 seconds (Warm-up)  
## 1.32 seconds (Sampling)  
## 3.103 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.755 seconds (Warm-up)  
## 1.314 seconds (Sampling)  
## 3.069 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.721 seconds (Warm-up)  
## 1.734 seconds (Sampling)  
## 3.455 seconds (Total)

Model Checks

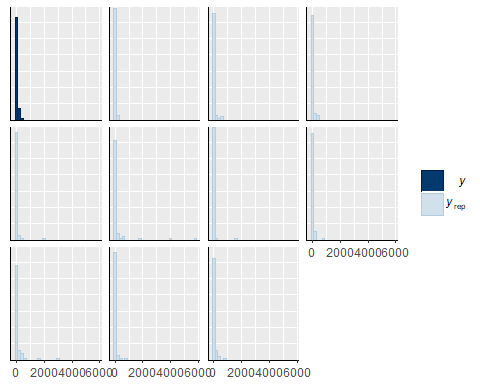
print(odolbio)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: odobio ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: larvbio2 (Number of observations: 71)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.40 0.32 0.01 1.18 2669 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## Intercept -0.44 1.05 -2.32 1.82 2012 1.00  
## date0f7D6D2016 4.14 1.09 1.94 6.19 2248 1.00  
## date0f8D2D2016 4.56 1.09 2.32 6.57 2284 1.00  
## trt0f2 -0.96 1.16 -3.16 1.36 3236 1.00  
## trt0f3 -1.00 1.16 -3.20 1.31 2839 1.00  
## trt0f4 -0.01 1.14 -2.27 2.22 2512 1.00  
## date0f7D6D2016:trt0f2 0.15 1.36 -2.43 2.83 4000 1.00  
## date0f8D2D2016:trt0f2 0.66 1.36 -1.98 3.46 4000 1.00  
## date0f7D6D2016:trt0f3 0.34 1.39 -2.27 3.13 4000 1.00  
## date0f8D2D2016:trt0f3 0.69 1.33 -1.84 3.34 3282 1.00  
## date0f7D6D2016:trt0f4 0.80 1.35 -1.79 3.55 4000 1.00  
## date0f8D2D2016:trt0f4 0.80 1.34 -1.72 3.45 2944 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 0.15 0.02 0.12 0.20 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(odolbio,type="hist")

## Using 10 posterior samples for ppc type 'hist' by default.

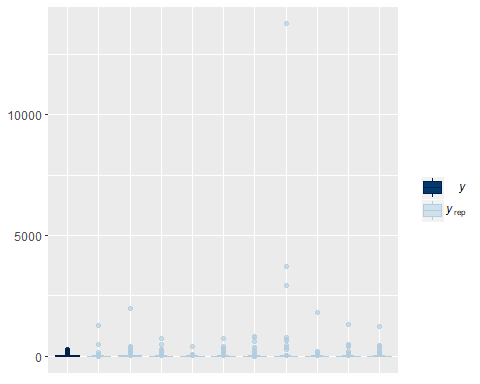
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(odolbio,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.

## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.



Example of Comparison among Treatments

lbpost<-posterior\_samples(lbio)  
str(lbpost)

## 'data.frame': 4000 obs. of 39 variables:  
## $ b\_Intercept : num 3.39 4.36 3.77 4.61 4.49 ...  
## $ b\_date0f7D6D2016 : num 1.4579 0.8825 0.7687 -0.1171 -0.0949 ...  
## $ b\_date0f8D2D2016 : num 1.954 0.3 0.498 -0.291 -0.124 ...  
## $ b\_trt0f2 : num 0.454 -0.524 -0.148 0.152 0.203 ...  
## $ b\_trt0f3 : num 0.473 -0.361 -0.184 -0.608 -0.741 ...  
## $ b\_trt0f4 : num 0.501 -0.22 0.246 -0.554 -0.587 ...  
## $ b\_date0f7D6D2016:trt0f2: num -1.6513 0.6795 0.288 0.0298 0.8369 ...  
## $ b\_date0f8D2D2016:trt0f2: num -2.243 0.324 0.431 0.564 0.94 ...  
## $ b\_date0f7D6D2016:trt0f3: num -1.4137 -0.89264 -0.93118 0.39127 -0.00747 ...  
## $ b\_date0f8D2D2016:trt0f3: num -1.5914 -0.0615 0.0836 -0.081 1.3411 ...  
## $ b\_date0f7D6D2016:trt0f4: num 0.21 0.548 1.152 2.077 2.051 ...  
## $ b\_date0f8D2D2016:trt0f4: num -0.705 -0.397 0.498 1.822 0.985 ...  
## $ sd\_tank\_\_Intercept : num 0.475 0.65 0.734 0.751 0.94 ...  
## $ shape : num 1.34 1.04 1.09 1.4 1.02 ...  
## $ r\_tank[1,Intercept] : num 0.0852 -0.8135 -0.647 -0.8826 -1.2368 ...  
## $ r\_tank[2,Intercept] : num 0.351 -1.094 -1.091 -1.087 -1.216 ...  
## $ r\_tank[3,Intercept] : num 0.114 0.31 0.303 0.224 0.352 ...  
## $ r\_tank[4,Intercept] : num 0.441 0.797 -0.351 1.071 0.708 ...  
## $ r\_tank[5,Intercept] : num -0.241 0.41 0.223 0.394 0.126 ...  
## $ r\_tank[6,Intercept] : num 0.3239 -0.7602 -0.0222 -1.3823 -0.9336 ...  
## $ r\_tank[7,Intercept] : num 0.3996 0.0501 -0.5622 -0.2823 -0.2245 ...  
## $ r\_tank[8,Intercept] : num -0.2082 -0.0552 -0.4557 0.1644 -0.359 ...  
## $ r\_tank[9,Intercept] : num -0.1763 0.4371 0.5393 -0.1478 -0.0418 ...  
## $ r\_tank[10,Intercept] : num -1.302 0.454 -0.925 -0.93 -2.296 ...  
## $ r\_tank[11,Intercept] : num 0.473 0.267 0.275 0.906 0.4 ...  
## $ r\_tank[12,Intercept] : num 0.313 -0.475 0.493 -0.324 0.321 ...  
## $ r\_tank[13,Intercept] : num 0.0162 -1.1736 -1.5226 -0.7869 -1.5272 ...  
## $ r\_tank[14,Intercept] : num -0.1905 -0.0604 -0.9884 0.4925 0.2336 ...  
## $ r\_tank[15,Intercept] : num -0.217 0.4863 0.0352 0.457 -0.9463 ...  
## $ r\_tank[16,Intercept] : num 1.043 -0.404 0.217 -0.371 -0.528 ...  
## $ r\_tank[17,Intercept] : num 0.46 0.328 1.234 0.2 1.613 ...  
## $ r\_tank[18,Intercept] : num -0.0393 -0.4577 -0.248 -1.9501 -1.5836 ...  
## $ r\_tank[19,Intercept] : num 0.294 -0.983 -0.456 -1.325 -0.749 ...  
## $ r\_tank[20,Intercept] : num -0.306 0.633 0.391 0.429 0.49 ...  
## $ r\_tank[21,Intercept] : num -0.472 0.411 -0.554 -1.131 -1.074 ...  
## $ r\_tank[22,Intercept] : num 0.0259 0.2593 0.8451 -0.3709 -0.6297 ...  
## $ r\_tank[23,Intercept] : num -0.0738 -1.1472 -0.872 -0.9576 -0.9445 ...  
## $ r\_tank[24,Intercept] : num 0.405 -0.945 -0.645 0.246 -0.251 ...  
## $ lp\_\_ : num -435 -429 -423 -434 -429 ...

#Date 7/6/16  
lbtrt1\_2<-lbpost$b\_Intercept+lbpost$b\_date0f7D6D2016  
lbtrt2\_2<-lbpost$b\_Intercept+lbpost$b\_trt0f2+lbpost$b\_date0f7D6D2016+lbpost$`b\_date0f7D6D2016:trt0f2`  
lbtrt3\_2<-lbpost$b\_Intercept+lbpost$b\_trt0f3+lbpost$b\_date0f7D6D2016+lbpost$`b\_date0f7D6D2016:trt0f3`  
lbtrt4\_2<-lbpost$b\_Intercept+lbpost$b\_trt0f4+lbpost$b\_date0f7D6D2016+lbpost$`b\_date0f7D6D2016:trt0f4`  
  
#Trt 1 & 4#  
lbdiff1\_4<-(exp(lbtrt4\_2)\*31.25)-(exp(lbtrt1\_2)\*31.25)  
mean(lbdiff1\_4)

## [1] 4518.632

quantile(lbdiff1\_4, probs=c(0.025,0.975))

## 2.5% 97.5%   
## -3015.079 17723.634

sum(lbdiff1\_4>0)/4000 #87% prob that difference >0

## [1] 0.872

Preparing to plot

Create data frame for each total larval biomass, chironomid larval biomass, and libellulid larval biomass, edit date as a properly ordered factor so that it plots dates in the right order, convert estimates and upper and lower quantiles to mgDM/m2.

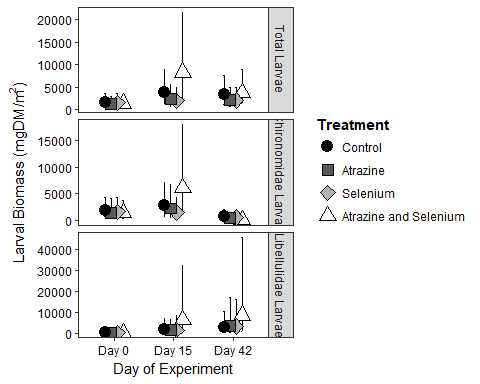
#TOTAL   
totlbplot<-marginal\_effects(lbio, robust=FALSE)  
totlbplot<-as.data.frame(totlbplot$`date0f:trt0f`)  
totlbplot$date2<-factor(totlbplot$date0f, levels = c("6/20/2016", "7/6/2016","8/2/2016"))  
totlbplot$est2<-(totlbplot$estimate\_\_)\*31.25  
totlbplot$upper2<-(totlbplot$upper\_\_)\*31.25  
totlbplot$lower2<-(totlbplot$lower\_\_)\*31.25  
  
#CHIRONOMID  
chirlbplot<-marginal\_effects(chirlbio, robust=FALSE)  
chirlbplot<-as.data.frame(chirlbplot$`date0f:trt0f`)  
chirlbplot$date2<-factor(chirlbplot$date0f, levels = c("6/20/2016", "7/6/2016","8/2/2016"))  
chirlbplot$est2<-(chirlbplot$estimate\_\_)\*31.25  
chirlbplot$upper2<-(chirlbplot$upper\_\_)\*31.25  
chirlbplot$lower2<-(chirlbplot$lower\_\_)\*31.25  
  
#LIBELLULID  
odolbplot<-marginal\_effects(odolbio, robust=FALSE)  
odolbplot<-as.data.frame(odolbplot$`date0f:trt0f`)  
odolbplot$date2<-factor(odolbplot$date0f, levels = c("6/20/2016", "7/6/2016","8/2/2016"))  
odolbplot$est2<-(odolbplot$estimate\_\_)\*31.25  
odolbplot$upper2<-(odolbplot$upper\_\_)\*31.25  
odolbplot$lower2<-(odolbplot$lower\_\_)\*31.25

Combine all three data frames to one, allowing for creation of a facetted plot, change date to day of experiment

chirlbplot$chirobiog<-NULL  
odolbplot$odobiog<-NULL  
totlbplot$totbiog<-NULL  
chirlbplot$chirobio<-NULL  
odolbplot$odobio<-NULL  
totlbplot$totbio<-NULL  
chirlbplot$group<-"Chironomidae Larvae"  
odolbplot$group<-"Libellulidae Larvae"  
totlbplot$group<-"Total Larvae"  
larv\_data<-rbind(chirlbplot,odolbplot,totlbplot)  
larv\_data$group2<-ifelse(larv\_data$group=="total\_larvae", "Total Larvae",  
 ifelse(larv\_data$group=="chiro\_larvae", "Chironomidae Larvae", "Libellulidae Larvae"))  
  
larv\_data$group3<-factor(larv\_data$group, levels=c("Total Larvae", "Chironomidae Larvae", "Libellulidae Larvae"))  
larv\_data$date3<-ifelse(larv\_data$date2=="6/20/2016", "Day 0",  
 ifelse(larv\_data$date2=="7/6/2016", "Day 15", "Day 42"))

Create plot

ggplot(larv\_data,aes(x=date3,y=est2, fill=trt0f, shape=trt0f))+  
 geom\_errorbar(aes(ymin=lower2, ymax=upper2), width=.1, position=position\_dodge(width=0.4))+  
 facet\_grid(group3~.,scales="free")+  
 geom\_point(position=position\_dodge(width=0.4), size=4)+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Larval Biomass (mgDM/m"^2,")")))+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_bw()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"),  
 panel.grid=element\_blank())



Larval Abundance

Larval abundace data come from the dbentab data file and are in #/0.032m2.

urlfile3<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/dbentab.csv"  
dbentab<-read.csv(urlfile3)  
dbentab$trt0f<-as.factor(dbentab$trt)  
dbentab$date0f<-as.factor(dbentab$date)

Model

get\_prior(nobugs~date0f+trt0f+date0f\*trt0f +(1|tank),data=dbentab,family=poisson(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D6D2016   
## 3 b date0f7D6D2016:trt0f2   
## 4 b date0f7D6D2016:trt0f3   
## 5 b date0f7D6D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, 4, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16

larvabund<-brm(nobugs~date0f+trt0f+date0f\*trt0f +(1|tank),data=dbentab,family=poisson(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,10),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 10.795 seconds (Warm-up)  
## 12.112 seconds (Sampling)  
## 22.907 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 10.377 seconds (Warm-up)  
## 11.109 seconds (Sampling)  
## 21.486 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 11.293 seconds (Warm-up)  
## 12.598 seconds (Sampling)  
## 23.891 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 11.203 seconds (Warm-up)  
## 10.028 seconds (Sampling)  
## 21.231 seconds (Total)

Model Checks

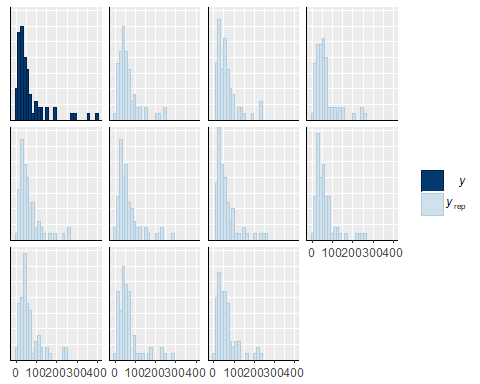
print(larvabund)

## Family: poisson   
## Links: mu = log   
## Formula: nobugs ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: dbentab (Number of observations: 72)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.68 0.11 0.50 0.94 1747 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## Intercept 4.07 0.29 3.51 4.66 1924 1.00  
## date0f7D6D2016 0.34 0.06 0.22 0.46 3103 1.00  
## date0f8D2D2016 -0.09 0.07 -0.22 0.05 3283 1.00  
## trt0f2 -0.22 0.41 -1.06 0.56 2104 1.00  
## trt0f3 -0.33 0.41 -1.12 0.51 2181 1.00  
## trt0f4 -0.39 0.41 -1.22 0.41 1976 1.00  
## date0f7D6D2016:trt0f2 -0.06 0.10 -0.25 0.13 4000 1.00  
## date0f8D2D2016:trt0f2 -0.21 0.11 -0.43 0.00 3466 1.00  
## date0f7D6D2016:trt0f3 -0.30 0.10 -0.50 -0.09 4000 1.00  
## date0f8D2D2016:trt0f3 -0.62 0.13 -0.86 -0.37 4000 1.00  
## date0f7D6D2016:trt0f4 0.80 0.09 0.62 0.99 3384 1.00  
## date0f8D2D2016:trt0f4 -0.40 0.12 -0.63 -0.18 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(larvabund,type="hist")

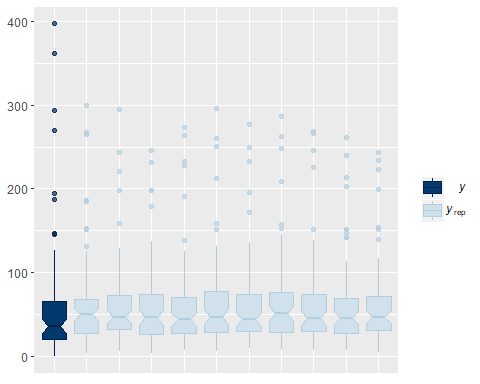
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(larvabund,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Example of Comparison among Treatments

mlpost<-posterior\_samples(larvabund)  
str(mlpost)

## 'data.frame': 4000 obs. of 38 variables:  
## $ b\_Intercept : num 4.25 4.28 4.31 4.43 4.34 ...  
## $ b\_date0f7D6D2016 : num 0.365 0.457 0.419 0.2 0.26 ...  
## $ b\_date0f8D2D2016 : num -0.1387 -0.0844 -0.1113 -0.1205 -0.1613 ...  
## $ b\_trt0f2 : num -0.618 -0.87 -0.906 -0.896 -0.779 ...  
## $ b\_trt0f3 : num -1.217 -1.05 -1.029 -0.908 -0.696 ...  
## $ b\_trt0f4 : num -0.277 -0.544 -0.666 -0.32 -0.369 ...  
## $ b\_date0f7D6D2016:trt0f2: num -0.0882 -0.0653 -0.2004 0.0912 -0.0168 ...  
## $ b\_date0f8D2D2016:trt0f2: num -0.14 -0.0292 -0.1336 -0.1639 -0.1955 ...  
## $ b\_date0f7D6D2016:trt0f3: num -0.537 -0.285 -0.36 -0.168 -0.2 ...  
## $ b\_date0f8D2D2016:trt0f3: num -0.654 -0.765 -0.662 -0.444 -0.536 ...  
## $ b\_date0f7D6D2016:trt0f4: num 0.797 0.675 0.716 0.864 0.823 ...  
## $ b\_date0f8D2D2016:trt0f4: num -0.418 -0.185 -0.119 -0.609 -0.522 ...  
## $ sd\_tank\_\_Intercept : num 0.774 0.746 0.786 0.659 0.632 ...  
## $ r\_tank[1,Intercept] : num -0.522 -0.726 -0.785 -0.782 -0.681 ...  
## $ r\_tank[2,Intercept] : num -0.44 -0.455 -0.423 -0.374 -0.697 ...  
## $ r\_tank[3,Intercept] : num 1.15 0.83 0.886 0.485 0.493 ...  
## $ r\_tank[4,Intercept] : num 0.366 0.187 0.36 0.31 0.259 ...  
## $ r\_tank[5,Intercept] : num 1.258 0.626 0.858 0.457 0.41 ...  
## $ r\_tank[6,Intercept] : num -0.867 -0.808 -0.947 -0.975 -0.912 ...  
## $ r\_tank[7,Intercept] : num 0.717 0.708 0.811 0.67 0.733 ...  
## $ r\_tank[8,Intercept] : num 0.253 0.376 -0.042 -0.309 -0.153 ...  
## $ r\_tank[9,Intercept] : num 0.395 0.691 0.528 0.397 0.388 ...  
## $ r\_tank[10,Intercept] : num -1.34 -0.914 -0.892 -1.098 -1.209 ...  
## $ r\_tank[11,Intercept] : num -0.134 -0.279 -0.365 -0.378 -0.203 ...  
## $ r\_tank[12,Intercept] : num 0.6598 0.1863 0.2233 -0.0775 -0.161 ...  
## $ r\_tank[13,Intercept] : num 0.0289 0.3314 0.454 0.2403 0.3105 ...  
## $ r\_tank[14,Intercept] : num -1.378 -0.999 -1.087 -1.347 -1.311 ...  
## $ r\_tank[15,Intercept] : num -0.227 -0.119 -0.226 -0.331 -0.315 ...  
## $ r\_tank[16,Intercept] : num 1 1.11 1.21 1.09 1.03 ...  
## $ r\_tank[17,Intercept] : num 0.746 0.784 0.715 0.736 0.758 ...  
## $ r\_tank[18,Intercept] : num 0.342 0.255 0.658 0.415 0.348 ...  
## $ r\_tank[19,Intercept] : num -0.397 -0.715 -0.652 -0.609 -0.463 ...  
## $ r\_tank[20,Intercept] : num 1.385 1.142 1.108 0.856 0.624 ...  
## $ r\_tank[21,Intercept] : num -0.3202 -0.1307 0.0469 -0.2832 0.0306 ...  
## $ r\_tank[22,Intercept] : num 0.38 0.622 0.586 0.308 0.421 ...  
## $ r\_tank[23,Intercept] : num 0.0797 -0.2912 -0.3454 -0.6936 -0.7311 ...  
## $ r\_tank[24,Intercept] : num 0.426 0.644 0.804 0.488 0.561 ...  
## $ lp\_\_ : num -1382 -1390 -1383 -1381 -1376 ...

ltrt1\_1<-mlpost$b\_Intercept  
ltrt2\_1<-mlpost$b\_Intercept+mlpost$b\_trt0f2  
  
#Differences on Date 1  
#Trt 1 & 2  
ltrt1\_2\_d1<-(exp(ltrt1\_1))-(exp(ltrt2\_1))  
mean(ltrt1\_2\_d1)

## [1] 12.03982

quantile(ltrt1\_2\_d1,probs=c(0.025,0.975))

## 2.5% 97.5%   
## -30.18286 62.51974

sum(ltrt1\_2\_d1>0)/4000 #72.3% prob diff > 0

## [1] 0.70275

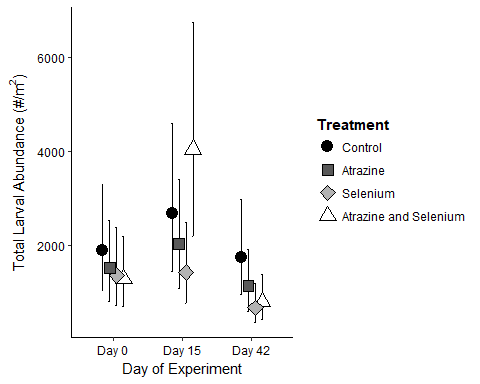
Preparing to plot

Create data frame, edit date as a properly ordered factor so that it plots dates in the right order, change date to day of experiment, convert estimates and upper and lower quantiles to #/m2.

laplot<-marginal\_effects(larvabund, robust=FALSE)  
laplot<-as.data.frame(laplot$`date0f:trt0f`)  
laplot$date2<-factor(laplot$date0f, levels = c("6/20/2016","7/6/2016","8/2/2016"))  
laplot$est2<-(laplot$estimate\_\_)\*31.25  
laplot$upper2<-(laplot$upper\_\_)\*31.25  
laplot$lower2<-(laplot$lower\_\_)\*31.25  
laplot$date3<-ifelse(laplot$date2=="6/20/2016", "Day 0",  
 ifelse(laplot$date2=="7/6/2016", "Day 15", "Day 42"))

Plot

ggplot(laplot,aes(x=date3,y=est2, fill=trt0f, shape=trt0f))+  
 geom\_errorbar(aes(ymin=lower2, ymax=upper2), width=.1, position=position\_dodge(width=0.4))+  
 geom\_point(position=position\_dodge(width=0.4), size=4)+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Total Larval Abundance (#/m"^2,")")))+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))



Chlorophyll

Data for chlorophyll models comes from the chla data file. Units for benthic chlorphyll are in ug/cm2 and units for water column chlorophyll are in mg/l.

urlfile4<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/chl.csv"  
chl<-read.csv(urlfile4)  
chl$trt0f<-as.factor(chl$trt)  
chl$date0f<-as.factor(chl$date)

Benthic Chlorophyll

Model

get\_prior(bchlug\_cm2\_new~date0f+trt0f+date0f\*trt0f +(1|tank),data=chl,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f6D28D2016:trt0f2   
## 4 b date0f6D28D2016:trt0f3   
## 5 b date0f6D28D2016:trt0f4   
## 6 b date0f7D12D2016   
## 7 b date0f7D12D2016:trt0f2   
## 8 b date0f7D12D2016:trt0f3   
## 9 b date0f7D12D2016:trt0f4   
## 10 b date0f7D18D2016   
## 11 b date0f7D18D2016:trt0f2   
## 12 b date0f7D18D2016:trt0f3   
## 13 b date0f7D18D2016:trt0f4   
## 14 b date0f7D26D2016   
## 15 b date0f7D26D2016:trt0f2   
## 16 b date0f7D26D2016:trt0f3   
## 17 b date0f7D26D2016:trt0f4   
## 18 b date0f7D4D2016   
## 19 b date0f7D4D2016:trt0f2   
## 20 b date0f7D4D2016:trt0f3   
## 21 b date0f7D4D2016:trt0f4   
## 22 b date0f8D2D2016   
## 23 b date0f8D2D2016:trt0f2   
## 24 b date0f8D2D2016:trt0f3   
## 25 b date0f8D2D2016:trt0f4   
## 26 b trt0f2   
## 27 b trt0f3   
## 28 b trt0f4   
## 29 student\_t(3, 2, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## 33 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33

bchlbrm<-brm(bchlug\_cm2\_new~date0f+trt0f+date0f\*trt0f +(1|tank),data=chl,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,5),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 7.399 seconds (Warm-up)  
## 6.082 seconds (Sampling)  
## 13.481 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.798 seconds (Warm-up)  
## 5.871 seconds (Sampling)  
## 12.669 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.15 seconds (Warm-up)  
## 6.013 seconds (Sampling)  
## 12.163 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.557 seconds (Warm-up)  
## 5.984 seconds (Sampling)  
## 12.541 seconds (Total)

Model Checks

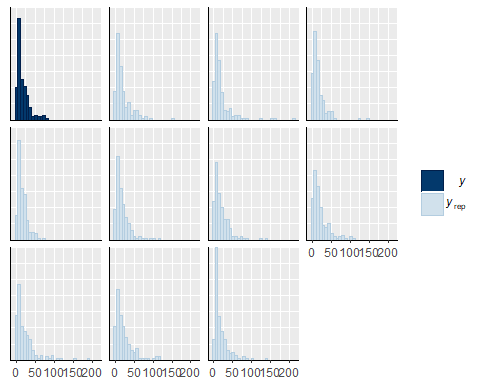
print(bchlbrm)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: bchlug\_cm2\_new ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: chl (Number of observations: 165)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.21 0.12 0.01 0.46 1178 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept 2.85 0.35 2.21 3.60 1054  
## date0f6D28D2016 0.03 0.47 -0.87 0.94 1424  
## date0f7D12D2016 0.37 0.47 -0.58 1.31 1468  
## date0f7D18D2016 0.19 0.47 -0.75 1.10 1360  
## date0f7D26D2016 0.60 0.47 -0.34 1.53 1325  
## date0f7D4D2016 -0.52 0.49 -1.50 0.45 1440  
## date0f8D2D2016 -0.46 0.51 -1.46 0.52 1642  
## trt0f2 0.46 0.49 -0.50 1.41 1296  
## trt0f3 0.45 0.49 -0.53 1.43 1254  
## trt0f4 0.43 0.49 -0.55 1.43 1290  
## date0f6D28D2016:trt0f2 -0.09 0.67 -1.37 1.24 1652  
## date0f7D12D2016:trt0f2 -0.86 0.67 -2.14 0.45 1682  
## date0f7D18D2016:trt0f2 -0.62 0.67 -1.93 0.74 1842  
## date0f7D26D2016:trt0f2 -1.08 0.67 -2.38 0.27 1681  
## date0f7D4D2016:trt0f2 0.31 0.69 -1.07 1.65 1646  
## date0f8D2D2016:trt0f2 0.05 0.71 -1.33 1.44 2075  
## date0f6D28D2016:trt0f3 -0.50 0.67 -1.85 0.82 1826  
## date0f7D12D2016:trt0f3 -1.52 0.67 -2.79 -0.14 1713  
## date0f7D18D2016:trt0f3 -1.34 0.68 -2.69 -0.03 1695  
## date0f7D26D2016:trt0f3 -0.87 0.67 -2.22 0.48 1685  
## date0f7D4D2016:trt0f3 0.24 0.69 -1.12 1.58 1830  
## date0f8D2D2016:trt0f3 -0.78 0.73 -2.20 0.65 1819  
## date0f6D28D2016:trt0f4 -0.54 0.69 -1.91 0.82 1732  
## date0f7D12D2016:trt0f4 -0.87 0.67 -2.14 0.48 1787  
## date0f7D18D2016:trt0f4 -0.69 0.66 -2.00 0.66 1551  
## date0f7D26D2016:trt0f4 -1.26 0.68 -2.57 0.05 1681  
## date0f7D4D2016:trt0f4 -0.37 0.67 -1.72 0.95 1630  
## date0f8D2D2016:trt0f4 -0.76 0.70 -2.13 0.62 1851  
## Rhat  
## Intercept 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016:trt0f2 1.00  
## date0f7D12D2016:trt0f2 1.00  
## date0f7D18D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f7D4D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f6D28D2016:trt0f3 1.00  
## date0f7D12D2016:trt0f3 1.00  
## date0f7D18D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f7D4D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f6D28D2016:trt0f4 1.00  
## date0f7D12D2016:trt0f4 1.00  
## date0f7D18D2016:trt0f4 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f7D4D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 1.48 0.17 1.18 1.83 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(bchlbrm,type="hist")

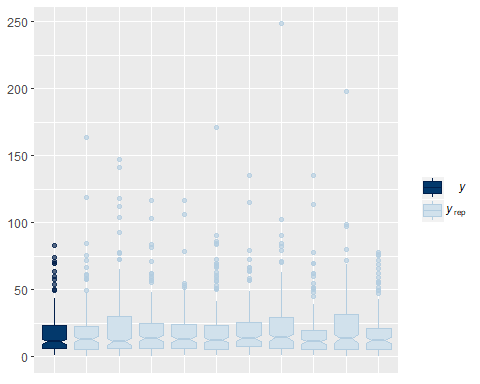
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(bchlbrm,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Example of Comparison among Treatments

chlpost<-posterior\_samples(bchlbrm)  
str(chlpost)

## 'data.frame': 4000 obs. of 55 variables:  
## $ b\_Intercept : num 2.22 2.89 3.08 3.2 2.89 ...  
## $ b\_date0f6D28D2016 : num 0.365 0.113 -0.338 0.201 -0.409 ...  
## $ b\_date0f7D12D2016 : num 1.0508 0.4649 0.3627 0.0335 -0.0468 ...  
## $ b\_date0f7D18D2016 : num 0.2596 0.4708 -0.3051 0.0406 -0.3371 ...  
## $ b\_date0f7D26D2016 : num 1.2058 0.4996 0.6877 -0.0899 0.8713 ...  
## $ b\_date0f7D4D2016 : num 0.805 -0.252 -1.039 -0.223 -0.708 ...  
## $ b\_date0f8D2D2016 : num 0.805 -1.12 0.189 -1.269 -0.248 ...  
## $ b\_trt0f2 : num 1.184 0.559 0.219 0.717 0.539 ...  
## $ b\_trt0f3 : num 0.843 1.599 -0.538 0.16 0.439 ...  
## $ b\_trt0f4 : num 0.6069 0.1415 0.9593 -0.2293 0.0752 ...  
## $ b\_date0f6D28D2016:trt0f2: num -0.873 -0.625 0.276 -0.62 -0.173 ...  
## $ b\_date0f7D12D2016:trt0f2: num -2.215 -1.218 -0.564 -0.905 -0.878 ...  
## $ b\_date0f7D18D2016:trt0f2: num -0.31 -0.945 -0.479 -1.54 -0.442 ...  
## $ b\_date0f7D26D2016:trt0f2: num -2.07 -1.01 -1.65 -1.08 -1.87 ...  
## $ b\_date0f7D4D2016:trt0f2 : num -1.3729 0.0768 0.7764 0.7002 0.1076 ...  
## $ b\_date0f8D2D2016:trt0f2 : num -1.658 0.447 -0.853 0.843 -0.686 ...  
## $ b\_date0f6D28D2016:trt0f3: num -0.3646 -1.9061 0.0911 -1.033 0.4433 ...  
## $ b\_date0f7D12D2016:trt0f3: num -1.842 -2.471 -0.809 -1.482 -1.695 ...  
## $ b\_date0f7D18D2016:trt0f3: num -0.628 -2.624 0.313 -1.046 -0.324 ...  
## $ b\_date0f7D26D2016:trt0f3: num -1.2105 -1.6162 -0.0397 -0.4707 -0.6866 ...  
## $ b\_date0f7D4D2016:trt0f3 : num -1.22 -1.557 1.205 -0.429 0.481 ...  
## $ b\_date0f8D2D2016:trt0f3 : num -1.6631 -1.1018 -0.1781 0.0902 -1.499 ...  
## $ b\_date0f6D28D2016:trt0f4: num 0.0939 -0.3439 -0.6825 -0.3153 1.0276 ...  
## $ b\_date0f7D12D2016:trt0f4: num -1.189 -0.476 -1.524 -0.178 -0.948 ...  
## $ b\_date0f7D18D2016:trt0f4: num -0.92792 -1.6133 -0.78548 0.00985 -0.43331 ...  
## $ b\_date0f7D26D2016:trt0f4: num -1.656 -0.711 -1.917 -0.13 -1.596 ...  
## $ b\_date0f7D4D2016:trt0f4 : num -1.273 -0.658 -0.693 -0.216 0.624 ...  
## $ b\_date0f8D2D2016:trt0f4 : num -1.768 0.211 -2.641 0.193 -0.804 ...  
## $ sd\_tank\_\_Intercept : num 0.078 0.119 0.216 0.284 0.285 ...  
## $ shape : num 1.11 1.8 1.43 1.52 1.34 ...  
## $ r\_tank[1,Intercept] : num -0.00701 0.06976 -0.09598 0.23199 0.0402 ...  
## $ r\_tank[2,Intercept] : num -0.0499 -0.1006 -0.1889 -0.363 0.069 ...  
## $ r\_tank[3,Intercept] : num -0.0973 -0.0159 -0.2933 -0.152 0.0215 ...  
## $ r\_tank[4,Intercept] : num 0.0545 -0.0596 0.0389 -0.1264 0.3958 ...  
## $ r\_tank[5,Intercept] : num -0.0245 -0.0583 -0.0509 -0.3135 0.0832 ...  
## $ r\_tank[6,Intercept] : num 0.0115 0.1336 0.1079 0.3002 0.3081 ...  
## $ r\_tank[7,Intercept] : num 0.18731 -0.21155 0.38361 -0.13865 -0.00561 ...  
## $ r\_tank[8,Intercept] : num -0.000672 -0.047304 0.015027 -0.217985 0.211586 ...  
## $ r\_tank[9,Intercept] : num -0.0324 0.0108 0.2497 -0.6202 0.2642 ...  
## $ r\_tank[10,Intercept] : num -0.121 0.236 0.154 -0.014 0.492 ...  
## $ r\_tank[11,Intercept] : num 0.1028 -0.1967 -0.0541 -0.2564 0.1874 ...  
## $ r\_tank[12,Intercept] : num -0.0658 0.1015 -0.1119 0.2569 -0.0143 ...  
## $ r\_tank[13,Intercept] : num 0.00705 0.08054 0.40712 -0.03702 0.53281 ...  
## $ r\_tank[14,Intercept] : num -0.0269 0.0157 -0.2389 0.2033 -0.0315 ...  
## $ r\_tank[15,Intercept] : num 0.0496 -0.0508 -0.0793 -0.1557 0.1398 ...  
## $ r\_tank[16,Intercept] : num 0.01926 -0.00738 0.02704 0.14891 -0.17559 ...  
## $ r\_tank[17,Intercept] : num -0.0851 0.1532 0.0789 0.1405 0.4741 ...  
## $ r\_tank[18,Intercept] : num 0.0529 -0.0549 0.4803 -0.3366 0.214 ...  
## $ r\_tank[19,Intercept] : num 0.0396 -0.081 0.1351 -0.2642 0.3171 ...  
## $ r\_tank[20,Intercept] : num -0.0287 0.2141 0.4431 0.0627 0.2061 ...  
## $ r\_tank[21,Intercept] : num 0.0253 0.0329 -0.3779 0.2399 -0.5491 ...  
## $ r\_tank[22,Intercept] : num 0.0703 -0.0031 -0.222 0.4919 -0.1683 ...  
## $ r\_tank[23,Intercept] : num -0.0757 0.1358 -0.0273 0.0231 0.0216 ...  
## $ r\_tank[24,Intercept] : num -0.0174 0.0567 0.0381 -0.1832 0.1288 ...  
## $ lp\_\_ : num -741 -738 -736 -737 -742 ...

bchltrt2\_3<-chlpost$b\_Intercept+chlpost$b\_date0f3+chlpost$b\_trt0f2+chlpost$`b\_date0f3:trt0f2`  
bchltrt4\_3<-chlpost$b\_Intercept+chlpost$b\_date0f3+chlpost$b\_trt0f4+chlpost$`b\_date0f3:trt0f4`  
  
#Date 3#  
#Trt 2 & 4#  
bchltrt2\_4\_d3<-exp(bchltrt2\_3)-exp(bchltrt4\_3)  
mean(bchltrt2\_4\_d3) #Average of 6.1 higher mg/l from control than selenium

## [1] NaN

quantile(bchltrt2\_4\_d3,probs=c(0.025,0.975))

## 2.5% 97.5%   
## NA NA

sum(bchltrt2\_4\_d3>0)/4000 #91.5% probability that these treatments are different

## [1] 0

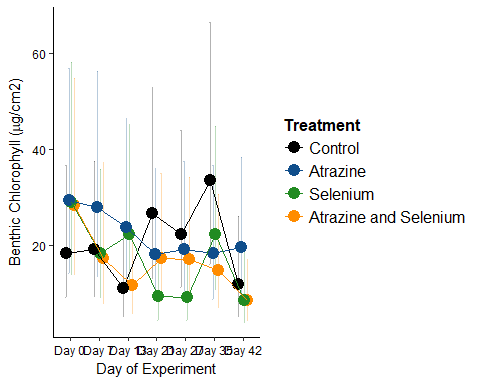
Preparing to plot

Create data frame, edit date as a properly ordered factor so that it plots dates in the right order, switch date to day of experiment.

bchlplot<-marginal\_effects(bchlbrm, robust=FALSE)  
bchlplot<-as.data.frame(bchlplot$`date0f:trt0f`)  
bchlplot$date2<-factor(bchlplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
  
bchlplot$date3<-ifelse(bchlplot$date2=="6/20/2016", "Day 0",  
 ifelse(bchlplot$date2=="6/28/2016", "Day 7",  
 ifelse(bchlplot$date2=="7/4/2016", "Day 13",  
 ifelse(bchlplot$date2=="7/12/2016", "Day 21",  
 ifelse(bchlplot$date2=="7/18/2016", "Day 27",  
 ifelse(bchlplot$date2=="7/26/2016", "Day 35", "Day 42"))))))  
  
bchlplot$date4<-factor(bchlplot$date3, levels=c("Day 0", "Day 7", "Day 13", "Day 21", "Day 27", "Day 35", "Day 42"))

Plot

ggplot(bchlplot,aes(x=date4,y=estimate\_\_, color=trt0f, group=trt0f))+  
 geom\_errorbar(aes(ymin=lower\_\_, ymax=upper\_\_), alpha=0.3, width=.1, position=position\_dodge(width=0.4))+  
 geom\_point(position=position\_dodge(width=0.4), size=4)+  
 geom\_line(position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Benthic Chlorophyll (", mu, "g/cm2)")))+  
 scale\_color\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0", "dodgerblue4","forestgreen","darkorange"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="12"))+  
 theme(legend.text = element\_text(size="12"))



Water Column Chlorophyll

Model

get\_prior(wchl~date0f+trt0f+date0f\*trt0f +(1|tank),data=chl,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f6D28D2016:trt0f2   
## 4 b date0f6D28D2016:trt0f3   
## 5 b date0f6D28D2016:trt0f4   
## 6 b date0f7D12D2016   
## 7 b date0f7D12D2016:trt0f2   
## 8 b date0f7D12D2016:trt0f3   
## 9 b date0f7D12D2016:trt0f4   
## 10 b date0f7D18D2016   
## 11 b date0f7D18D2016:trt0f2   
## 12 b date0f7D18D2016:trt0f3   
## 13 b date0f7D18D2016:trt0f4   
## 14 b date0f7D26D2016   
## 15 b date0f7D26D2016:trt0f2   
## 16 b date0f7D26D2016:trt0f3   
## 17 b date0f7D26D2016:trt0f4   
## 18 b date0f7D4D2016   
## 19 b date0f7D4D2016:trt0f2   
## 20 b date0f7D4D2016:trt0f3   
## 21 b date0f7D4D2016:trt0f4   
## 22 b date0f8D2D2016   
## 23 b date0f8D2D2016:trt0f2   
## 24 b date0f8D2D2016:trt0f3   
## 25 b date0f8D2D2016:trt0f4   
## 26 b trt0f2   
## 27 b trt0f3   
## 28 b trt0f4   
## 29 student\_t(3, 1, 10) Intercept   
## 30 student\_t(3, 0, 10) sd   
## 31 sd tank   
## 32 sd Intercept tank   
## 33 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33

wchlbrm<-brm(wchl~date0f+trt0f+date0f\*trt0f +(1|tank),data=chl,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,5),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 7.376 seconds (Warm-up)  
## 6.346 seconds (Sampling)  
## 13.722 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 7.309 seconds (Warm-up)  
## 6.405 seconds (Sampling)  
## 13.714 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 8.171 seconds (Warm-up)  
## 6.048 seconds (Sampling)  
## 14.219 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.606 seconds (Warm-up)  
## 6.316 seconds (Sampling)  
## 12.922 seconds (Total)

Model Checks

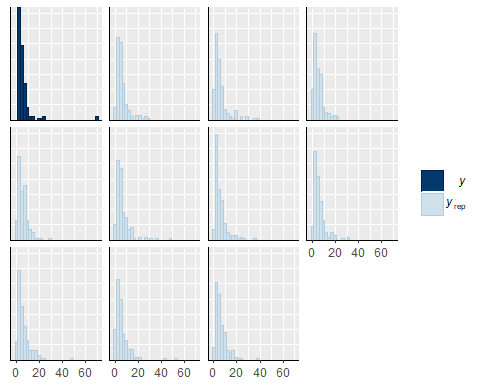
print(wchlbrm)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: wchl ~ date0f + trt0f + date0f \* trt0f + (1 | tank)   
## Data: chl (Number of observations: 165)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.28 0.09 0.10 0.46 1452 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept 1.76 0.27 1.26 2.31 1181  
## date0f6D28D2016 0.25 0.34 -0.43 0.91 1567  
## date0f7D12D2016 -0.70 0.35 -1.37 -0.02 1386  
## date0f7D18D2016 -0.42 0.34 -1.10 0.22 1447  
## date0f7D26D2016 -0.38 0.35 -1.09 0.30 1562  
## date0f7D4D2016 -0.51 0.34 -1.17 0.16 1537  
## date0f8D2D2016 -0.10 0.36 -0.79 0.62 1567  
## trt0f2 -0.27 0.39 -1.02 0.48 1480  
## trt0f3 -0.56 0.38 -1.28 0.19 1391  
## trt0f4 -0.01 0.39 -0.78 0.76 1370  
## date0f6D28D2016:trt0f2 0.77 0.51 -0.24 1.79 1967  
## date0f7D12D2016:trt0f2 1.21 0.49 0.24 2.15 1985  
## date0f7D18D2016:trt0f2 0.92 0.49 -0.06 1.88 1818  
## date0f7D26D2016:trt0f2 1.17 0.50 0.19 2.19 1920  
## date0f7D4D2016:trt0f2 1.44 0.51 0.44 2.45 1923  
## date0f8D2D2016:trt0f2 0.27 0.50 -0.73 1.25 1879  
## date0f6D28D2016:trt0f3 0.20 0.49 -0.76 1.13 1794  
## date0f7D12D2016:trt0f3 0.49 0.49 -0.47 1.45 1665  
## date0f7D18D2016:trt0f3 0.59 0.48 -0.33 1.52 1671  
## date0f7D26D2016:trt0f3 0.56 0.49 -0.40 1.49 1608  
## date0f7D4D2016:trt0f3 0.67 0.48 -0.30 1.60 1826  
## date0f8D2D2016:trt0f3 -0.02 0.50 -1.00 0.95 1846  
## date0f6D28D2016:trt0f4 -0.41 0.52 -1.41 0.61 1766  
## date0f7D12D2016:trt0f4 1.18 0.50 0.17 2.20 1754  
## date0f7D18D2016:trt0f4 0.40 0.50 -0.58 1.38 1523  
## date0f7D26D2016:trt0f4 0.47 0.51 -0.51 1.49 1632  
## date0f7D4D2016:trt0f4 0.02 0.49 -0.96 0.96 1561  
## date0f8D2D2016:trt0f4 -0.12 0.51 -1.13 0.86 1703  
## Rhat  
## Intercept 1.00  
## date0f6D28D2016 1.00  
## date0f7D12D2016 1.00  
## date0f7D18D2016 1.00  
## date0f7D26D2016 1.00  
## date0f7D4D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f6D28D2016:trt0f2 1.00  
## date0f7D12D2016:trt0f2 1.00  
## date0f7D18D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f7D4D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f6D28D2016:trt0f3 1.00  
## date0f7D12D2016:trt0f3 1.00  
## date0f7D18D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f7D4D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f6D28D2016:trt0f4 1.00  
## date0f7D12D2016:trt0f4 1.00  
## date0f7D18D2016:trt0f4 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f7D4D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 2.85 0.35 2.20 3.57 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(wchlbrm,type="hist")

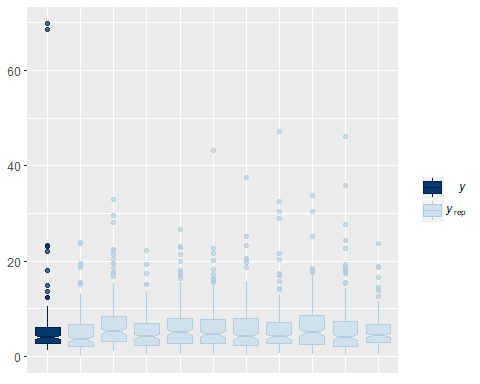
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(wchlbrm,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Example of Comparison among Treatments

wchlpost<-posterior\_samples(wchlbrm)  
str(wchlpost)

## 'data.frame': 4000 obs. of 55 variables:  
## $ b\_Intercept : num 1.68 1.66 1.83 1.7 1.3 ...  
## $ b\_date0f6D28D2016 : num 0.0659 0.0491 -0.1959 0.5249 0.1349 ...  
## $ b\_date0f7D12D2016 : num -0.592 -1.279 -1.076 -0.527 -0.694 ...  
## $ b\_date0f7D18D2016 : num -0.481 -0.412 -0.381 -0.289 -0.352 ...  
## $ b\_date0f7D26D2016 : num -0.634 -0.843 -1.043 -0.155 -0.152 ...  
## $ b\_date0f7D4D2016 : num -0.5248 -0.4089 -0.6617 -0.5016 0.0198 ...  
## $ b\_date0f8D2D2016 : num -0.308 0.093 0.144 0.185 0.153 ...  
## $ b\_trt0f2 : num -0.488 -0.438 -0.302 -0.843 0.686 ...  
## $ b\_trt0f3 : num -0.415 -0.697 -1.142 -0.387 -0.247 ...  
## $ b\_trt0f4 : num 0.26 0.496 0.141 0.4 0.108 ...  
## $ b\_date0f6D28D2016:trt0f2: num 1.062 1.251 1.136 0.387 0.46 ...  
## $ b\_date0f7D12D2016:trt0f2: num 1.277 1.86 1.699 1.528 0.785 ...  
## $ b\_date0f7D18D2016:trt0f2: num 0.931 1.109 1.182 1.467 0.211 ...  
## $ b\_date0f7D26D2016:trt0f2: num 1.427 1.53 1.803 1.629 0.956 ...  
## $ b\_date0f7D4D2016:trt0f2 : num 1.134 1.747 1.668 1.915 0.803 ...  
## $ b\_date0f8D2D2016:trt0f2 : num 0.4838 -0.0387 -0.1503 0.1691 -0.867 ...  
## $ b\_date0f6D28D2016:trt0f3: num 0.24 1.01 1.215 0.236 0.622 ...  
## $ b\_date0f7D12D2016:trt0f3: num 0.723 1.065 0.959 0.236 0.231 ...  
## $ b\_date0f7D18D2016:trt0f3: num 0.772 0.939 0.864 0.89 0.295 ...  
## $ b\_date0f7D26D2016:trt0f3: num 0.9413 1.5231 1.2353 -0.0477 0.1253 ...  
## $ b\_date0f7D4D2016:trt0f3 : num 0.815 0.649 0.797 0.898 -0.175 ...  
## $ b\_date0f8D2D2016:trt0f3 : num -0.255 0.568 0.82 0.187 0.557 ...  
## $ b\_date0f6D28D2016:trt0f4: num -0.406 -0.483 -0.429 -0.968 0.137 ...  
## $ b\_date0f7D12D2016:trt0f4: num 0.717 1.195 1.125 0.622 1.451 ...  
## $ b\_date0f7D18D2016:trt0f4: num 0.543 -0.178 -0.491 0.036 0.447 ...  
## $ b\_date0f7D26D2016:trt0f4: num 0.417 0.579 0.481 -0.445 0.938 ...  
## $ b\_date0f7D4D2016:trt0f4 : num -0.355 -0.408 -0.507 0.147 -0.139 ...  
## $ b\_date0f8D2D2016:trt0f4 : num -0.136 -1.101 -0.89 -1.229 -0.212 ...  
## $ sd\_tank\_\_Intercept : num 0.323 0.293 0.312 0.3 0.236 ...  
## $ shape : num 2.91 1.84 1.96 3.16 2.66 ...  
## $ r\_tank[1,Intercept] : num 0.1044 -0.2996 -0.3111 -0.0315 -0.1767 ...  
## $ r\_tank[2,Intercept] : num 0.1138 -0.0973 -0.1052 -0.2176 -0.1083 ...  
## $ r\_tank[3,Intercept] : num -0.0484 0.0395 0.2413 -0.1036 0.142 ...  
## $ r\_tank[4,Intercept] : num 0.0122 0.137 0.0687 -0.1944 0.0863 ...  
## $ r\_tank[5,Intercept] : num 0.0883 0.3292 0.4489 0.0108 0.2227 ...  
## $ r\_tank[6,Intercept] : num 0.0447 0.2323 0.2365 0.0197 0.2851 ...  
## $ r\_tank[7,Intercept] : num -0.166 0.163 0.163 -0.281 0.143 ...  
## $ r\_tank[8,Intercept] : num 0.0481 0.4854 0.4318 -0.4853 0.4302 ...  
## $ r\_tank[9,Intercept] : num -0.1179 0.2166 0.2206 -0.3147 0.0319 ...  
## $ r\_tank[10,Intercept] : num 0.1556 0.1516 0.0213 0.5839 -0.1429 ...  
## $ r\_tank[11,Intercept] : num 0.2738 0.1256 0.2467 0.0206 0.2951 ...  
## $ r\_tank[12,Intercept] : num 0.0352 -0.2143 0.0291 -0.1172 -0.0118 ...  
## $ r\_tank[13,Intercept] : num -0.212 -0.106 -0.144 -0.358 -0.113 ...  
## $ r\_tank[14,Intercept] : num -0.4524 -0.0133 -0.0818 -0.2548 -0.2194 ...  
## $ r\_tank[15,Intercept] : num 0.05908 0.02313 0.05342 0.00766 0.04362 ...  
## $ r\_tank[16,Intercept] : num 0.0268 -0.0188 0.1461 -0.0196 0.1008 ...  
## $ r\_tank[17,Intercept] : num 0.0058 0.0742 -0.0392 -0.0842 -0.0454 ...  
## $ r\_tank[18,Intercept] : num 0.3946 0.2503 0.202 0.4051 0.0111 ...  
## $ r\_tank[19,Intercept] : num 0.2719 0.6664 0.7322 0.0506 0.6087 ...  
## $ r\_tank[20,Intercept] : num -0.173 -0.479 -0.267 -0.249 -0.217 ...  
## $ r\_tank[21,Intercept] : num 0.614 0.15 0.305 0.675 0.188 ...  
## $ r\_tank[22,Intercept] : num 0.00794 0.47811 0.52784 -0.02879 0.36576 ...  
## $ r\_tank[23,Intercept] : num -0.4876 -0.0145 0.2508 -0.3526 0.1865 ...  
## $ r\_tank[24,Intercept] : num 0.3999 -0.0081 -0.1136 -0.176 -0.2406 ...  
## $ lp\_\_ : num -499 -519 -519 -509 -518 ...

wchltrt1\_1<-wchlpost$b\_Intercept  
wchltrt3\_1<-wchlpost$b\_Intercept+wchlpost$b\_trt0f3  
  
#Date 1#  
#Trt 1 & 3#  
wchltrt1\_3\_d1<-exp(wchltrt1\_1)-exp(wchltrt3\_1)  
mean(wchltrt1\_3\_d1) #Average of 2.7 higher mg/l from control than selenium

## [1] 2.57173

quantile(wchltrt1\_3\_d1,probs=c(0.025,0.975))

## 2.5% 97.5%   
## -0.8603267 6.7904023

sum(wchltrt1\_3\_d1>0)/4000 #93% probability that these treatments are different

## [1] 0.93125

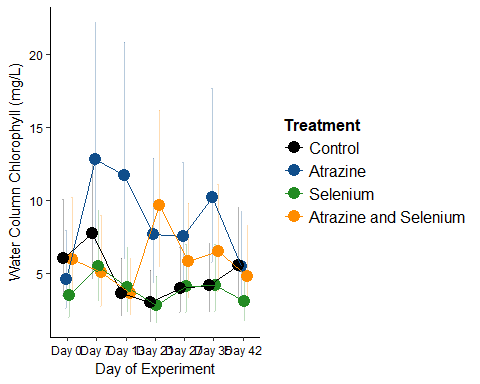
Preparing to plot

Create data frame, edit date as a properly ordered factor so that it plots dates in the right order, change date to day of experiment

wchlplot<-marginal\_effects(wchlbrm, robust=FALSE)  
wchlplot<-as.data.frame(wchlplot$`date0f:trt0f`)  
wchlplot$date2<-factor(wchlplot$date0f, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
  
wchlplot$date3<-ifelse(wchlplot$date2=="6/20/2016", "Day 0",  
 ifelse(wchlplot$date2=="6/28/2016", "Day 7",  
 ifelse(wchlplot$date2=="7/4/2016", "Day 13",  
 ifelse(wchlplot$date2=="7/12/2016", "Day 21",  
 ifelse(wchlplot$date2=="7/18/2016", "Day 27",  
 ifelse(wchlplot$date2=="7/26/2016", "Day 35", "Day 42"))))))  
  
wchlplot$date4<-factor(wchlplot$date3, levels=c("Day 0", "Day 7", "Day 13", "Day 21", "Day 27", "Day 35", "Day 42"))

Plot

ggplot(wchlplot,aes(x=date4,y=estimate\_\_, color=trt0f,group=trt0f))+  
 geom\_errorbar(aes(ymin=lower\_\_, ymax=upper\_\_),alpha=0.3, width=.1, position=position\_dodge(width=0.4))+  
 geom\_point(position=position\_dodge(width=0.4), size=4)+  
 geom\_line(position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab("Water Column Chlorophyll (mg/L)")+  
 scale\_color\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0", "dodgerblue4","forestgreen","darkorange"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="12"))+  
 theme(legend.text = element\_text(size="12"))



Body Size Models

Data

urlfile5<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/chiro\_g\_ind.csv"  
chiro\_g\_ind<-read.csv(urlfile5)  
chiro\_g\_ind$trt0f<-as.factor(chiro\_g\_ind$trt)  
chiro\_g\_ind$date0f<-as.factor(chiro\_g\_ind$date)  
chiro\_g\_ind$morph0f<-as.factor(chiro\_g\_ind$morph)  
  
urlfile6<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/dragon\_g\_ind.csv"  
dragon\_g\_ind<-read.csv(urlfile6)  
dragon\_g\_ind$trt0f<-as.factor(dragon\_g\_ind$trt)  
dragon\_g\_ind$date0f<-as.factor(dragon\_g\_ind$date)  
  
urlfile7<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/may\_g\_ind.csv"  
may\_g\_ind<-read.csv(urlfile7)  
may\_g\_ind$trt0f<-as.factor(may\_g\_ind$trt)  
may\_g\_ind$date0f<-as.factor(may\_g\_ind$date)

Effect of Treatment on Chironomid Body Size

get\_prior(g\_ind~date0f\*trt0f\*morph0f+(1|tank),data=chiro\_g\_ind,family=Gamma(link="log"))

## prior class coef group  
## 1 b   
## 2 b date0f6D28D2016   
## 3 b date0f6D28D2016:morph0fB   
## 4 b date0f6D28D2016:morph0fC   
## 5 b date0f6D28D2016:morph0fD   
## 6 b date0f6D28D2016:morph0fE   
## 7 b date0f6D28D2016:morph0fF   
## 8 b date0f6D28D2016:trt0f2   
## 9 b date0f6D28D2016:trt0f2:morph0fB   
## 10 b date0f6D28D2016:trt0f2:morph0fC   
## 11 b date0f6D28D2016:trt0f2:morph0fD   
## 12 b date0f6D28D2016:trt0f2:morph0fE   
## 13 b date0f6D28D2016:trt0f2:morph0fF   
## 14 b date0f6D28D2016:trt0f3   
## 15 b date0f6D28D2016:trt0f3:morph0fB   
## 16 b date0f6D28D2016:trt0f3:morph0fC   
## 17 b date0f6D28D2016:trt0f3:morph0fD   
## 18 b date0f6D28D2016:trt0f3:morph0fE   
## 19 b date0f6D28D2016:trt0f3:morph0fF   
## 20 b date0f6D28D2016:trt0f4   
## 21 b date0f6D28D2016:trt0f4:morph0fB   
## 22 b date0f6D28D2016:trt0f4:morph0fC   
## 23 b date0f6D28D2016:trt0f4:morph0fD   
## 24 b date0f6D28D2016:trt0f4:morph0fE   
## 25 b date0f6D28D2016:trt0f4:morph0fF   
## 26 b date0f7D12D2016   
## 27 b date0f7D12D2016:morph0fB   
## 28 b date0f7D12D2016:morph0fC   
## 29 b date0f7D12D2016:morph0fD   
## 30 b date0f7D12D2016:morph0fE   
## 31 b date0f7D12D2016:morph0fF   
## 32 b date0f7D12D2016:trt0f2   
## 33 b date0f7D12D2016:trt0f2:morph0fB   
## 34 b date0f7D12D2016:trt0f2:morph0fC   
## 35 b date0f7D12D2016:trt0f2:morph0fD   
## 36 b date0f7D12D2016:trt0f2:morph0fE   
## 37 b date0f7D12D2016:trt0f2:morph0fF   
## 38 b date0f7D12D2016:trt0f3   
## 39 b date0f7D12D2016:trt0f3:morph0fB   
## 40 b date0f7D12D2016:trt0f3:morph0fC   
## 41 b date0f7D12D2016:trt0f3:morph0fD   
## 42 b date0f7D12D2016:trt0f3:morph0fE   
## 43 b date0f7D12D2016:trt0f3:morph0fF   
## 44 b date0f7D12D2016:trt0f4   
## 45 b date0f7D12D2016:trt0f4:morph0fB   
## 46 b date0f7D12D2016:trt0f4:morph0fC   
## 47 b date0f7D12D2016:trt0f4:morph0fD   
## 48 b date0f7D12D2016:trt0f4:morph0fE   
## 49 b date0f7D12D2016:trt0f4:morph0fF   
## 50 b date0f7D18D2016   
## 51 b date0f7D18D2016:morph0fB   
## 52 b date0f7D18D2016:morph0fC   
## 53 b date0f7D18D2016:morph0fD   
## 54 b date0f7D18D2016:morph0fE   
## 55 b date0f7D18D2016:morph0fF   
## 56 b date0f7D18D2016:trt0f2   
## 57 b date0f7D18D2016:trt0f2:morph0fB   
## 58 b date0f7D18D2016:trt0f2:morph0fC   
## 59 b date0f7D18D2016:trt0f2:morph0fD   
## 60 b date0f7D18D2016:trt0f2:morph0fE   
## 61 b date0f7D18D2016:trt0f2:morph0fF   
## 62 b date0f7D18D2016:trt0f3   
## 63 b date0f7D18D2016:trt0f3:morph0fB   
## 64 b date0f7D18D2016:trt0f3:morph0fC   
## 65 b date0f7D18D2016:trt0f3:morph0fD   
## 66 b date0f7D18D2016:trt0f3:morph0fE   
## 67 b date0f7D18D2016:trt0f3:morph0fF   
## 68 b date0f7D18D2016:trt0f4   
## 69 b date0f7D18D2016:trt0f4:morph0fB   
## 70 b date0f7D18D2016:trt0f4:morph0fC   
## 71 b date0f7D18D2016:trt0f4:morph0fD   
## 72 b date0f7D18D2016:trt0f4:morph0fE   
## 73 b date0f7D18D2016:trt0f4:morph0fF   
## 74 b date0f7D26D2016   
## 75 b date0f7D26D2016:morph0fB   
## 76 b date0f7D26D2016:morph0fC   
## 77 b date0f7D26D2016:morph0fD   
## 78 b date0f7D26D2016:morph0fE   
## 79 b date0f7D26D2016:morph0fF   
## 80 b date0f7D26D2016:trt0f2   
## 81 b date0f7D26D2016:trt0f2:morph0fB   
## 82 b date0f7D26D2016:trt0f2:morph0fC   
## 83 b date0f7D26D2016:trt0f2:morph0fD   
## 84 b date0f7D26D2016:trt0f2:morph0fE   
## 85 b date0f7D26D2016:trt0f2:morph0fF   
## 86 b date0f7D26D2016:trt0f3   
## 87 b date0f7D26D2016:trt0f3:morph0fB   
## 88 b date0f7D26D2016:trt0f3:morph0fC   
## 89 b date0f7D26D2016:trt0f3:morph0fD   
## 90 b date0f7D26D2016:trt0f3:morph0fE   
## 91 b date0f7D26D2016:trt0f3:morph0fF   
## 92 b date0f7D26D2016:trt0f4   
## 93 b date0f7D26D2016:trt0f4:morph0fB   
## 94 b date0f7D26D2016:trt0f4:morph0fC   
## 95 b date0f7D26D2016:trt0f4:morph0fD   
## 96 b date0f7D26D2016:trt0f4:morph0fE   
## 97 b date0f7D26D2016:trt0f4:morph0fF   
## 98 b date0f7D4D2016   
## 99 b date0f7D4D2016:morph0fB   
## 100 b date0f7D4D2016:morph0fC   
## 101 b date0f7D4D2016:morph0fD   
## 102 b date0f7D4D2016:morph0fE   
## 103 b date0f7D4D2016:morph0fF   
## 104 b date0f7D4D2016:trt0f2   
## 105 b date0f7D4D2016:trt0f2:morph0fB   
## 106 b date0f7D4D2016:trt0f2:morph0fC   
## 107 b date0f7D4D2016:trt0f2:morph0fD   
## 108 b date0f7D4D2016:trt0f2:morph0fE   
## 109 b date0f7D4D2016:trt0f2:morph0fF   
## 110 b date0f7D4D2016:trt0f3   
## 111 b date0f7D4D2016:trt0f3:morph0fB   
## 112 b date0f7D4D2016:trt0f3:morph0fC   
## 113 b date0f7D4D2016:trt0f3:morph0fD   
## 114 b date0f7D4D2016:trt0f3:morph0fE   
## 115 b date0f7D4D2016:trt0f3:morph0fF   
## 116 b date0f7D4D2016:trt0f4   
## 117 b date0f7D4D2016:trt0f4:morph0fB   
## 118 b date0f7D4D2016:trt0f4:morph0fC   
## 119 b date0f7D4D2016:trt0f4:morph0fD   
## 120 b date0f7D4D2016:trt0f4:morph0fE   
## 121 b date0f7D4D2016:trt0f4:morph0fF   
## 122 b date0f8D2D2016   
## 123 b date0f8D2D2016:morph0fB   
## 124 b date0f8D2D2016:morph0fC   
## 125 b date0f8D2D2016:morph0fD   
## 126 b date0f8D2D2016:morph0fE   
## 127 b date0f8D2D2016:morph0fF   
## 128 b date0f8D2D2016:trt0f2   
## 129 b date0f8D2D2016:trt0f2:morph0fB   
## 130 b date0f8D2D2016:trt0f2:morph0fC   
## 131 b date0f8D2D2016:trt0f2:morph0fD   
## 132 b date0f8D2D2016:trt0f2:morph0fE   
## 133 b date0f8D2D2016:trt0f2:morph0fF   
## 134 b date0f8D2D2016:trt0f3   
## 135 b date0f8D2D2016:trt0f3:morph0fB   
## 136 b date0f8D2D2016:trt0f3:morph0fC   
## 137 b date0f8D2D2016:trt0f3:morph0fD   
## 138 b date0f8D2D2016:trt0f3:morph0fE   
## 139 b date0f8D2D2016:trt0f3:morph0fF   
## 140 b date0f8D2D2016:trt0f4   
## 141 b date0f8D2D2016:trt0f4:morph0fB   
## 142 b date0f8D2D2016:trt0f4:morph0fC   
## 143 b date0f8D2D2016:trt0f4:morph0fD   
## 144 b date0f8D2D2016:trt0f4:morph0fE   
## 145 b date0f8D2D2016:trt0f4:morph0fF   
## 146 b morph0fB   
## 147 b morph0fC   
## 148 b morph0fD   
## 149 b morph0fE   
## 150 b morph0fF   
## 151 b trt0f2   
## 152 b trt0f2:morph0fB   
## 153 b trt0f2:morph0fC   
## 154 b trt0f2:morph0fD   
## 155 b trt0f2:morph0fE   
## 156 b trt0f2:morph0fF   
## 157 b trt0f3   
## 158 b trt0f3:morph0fB   
## 159 b trt0f3:morph0fC   
## 160 b trt0f3:morph0fD   
## 161 b trt0f3:morph0fE   
## 162 b trt0f3:morph0fF   
## 163 b trt0f4   
## 164 b trt0f4:morph0fB   
## 165 b trt0f4:morph0fC   
## 166 b trt0f4:morph0fD   
## 167 b trt0f4:morph0fE   
## 168 b trt0f4:morph0fF   
## 169 student\_t(3, -9, 10) Intercept   
## 170 student\_t(3, 0, 10) sd   
## 171 sd tank  
## 172 sd Intercept tank  
## 173 gamma(0.01, 0.01) shape   
## resp dpar nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25   
## 26   
## 27   
## 28   
## 29   
## 30   
## 31   
## 32   
## 33   
## 34   
## 35   
## 36   
## 37   
## 38   
## 39   
## 40   
## 41   
## 42   
## 43   
## 44   
## 45   
## 46   
## 47   
## 48   
## 49   
## 50   
## 51   
## 52   
## 53   
## 54   
## 55   
## 56   
## 57   
## 58   
## 59   
## 60   
## 61   
## 62   
## 63   
## 64   
## 65   
## 66   
## 67   
## 68   
## 69   
## 70   
## 71   
## 72   
## 73   
## 74   
## 75   
## 76   
## 77   
## 78   
## 79   
## 80   
## 81   
## 82   
## 83   
## 84   
## 85   
## 86   
## 87   
## 88   
## 89   
## 90   
## 91   
## 92   
## 93   
## 94   
## 95   
## 96   
## 97   
## 98   
## 99   
## 100   
## 101   
## 102   
## 103   
## 104   
## 105   
## 106   
## 107   
## 108   
## 109   
## 110   
## 111   
## 112   
## 113   
## 114   
## 115   
## 116   
## 117   
## 118   
## 119   
## 120   
## 121   
## 122   
## 123   
## 124   
## 125   
## 126   
## 127   
## 128   
## 129   
## 130   
## 131   
## 132   
## 133   
## 134   
## 135   
## 136   
## 137   
## 138   
## 139   
## 140   
## 141   
## 142   
## 143   
## 144   
## 145   
## 146   
## 147   
## 148   
## 149   
## 150   
## 151   
## 152   
## 153   
## 154   
## 155   
## 156   
## 157   
## 158   
## 159   
## 160   
## 161   
## 162   
## 163   
## 164   
## 165   
## 166   
## 167   
## 168   
## 169   
## 170   
## 171   
## 172   
## 173

chirobody<-brm(g\_ind~date0f\*trt0f\*morph0f+(1|tank),data=chiro\_g\_ind,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,1),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0.002 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 77.148 seconds (Warm-up)  
## 97.935 seconds (Sampling)  
## 175.083 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0.002 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 77.328 seconds (Warm-up)  
## 90.168 seconds (Sampling)  
## 167.496 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 92.285 seconds (Warm-up)  
## 90.405 seconds (Sampling)  
## 182.69 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0.002 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 81.175 seconds (Warm-up)  
## 85.771 seconds (Sampling)  
## 166.946 seconds (Total)

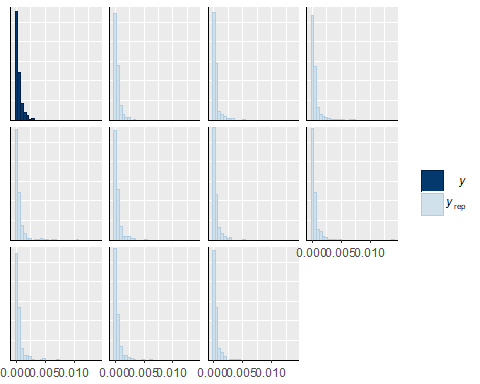
print(chirobody)

## Family: gamma   
## Links: mu = log; shape = identity   
## Formula: g\_ind ~ date0f \* trt0f \* morph0f + (1 | tank)   
## Data: chiro\_g\_ind (Number of observations: 477)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.21 0.06 0.11 0.34 1710 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI  
## Intercept -7.13 0.22 -7.56 -6.69  
## date0f6D28D2016 0.19 0.41 -0.58 1.01  
## date0f7D12D2016 0.31 0.28 -0.24 0.85  
## date0f7D18D2016 -0.09 0.27 -0.61 0.46  
## date0f7D26D2016 0.18 0.26 -0.32 0.68  
## date0f7D4D2016 0.36 0.31 -0.22 0.98  
## date0f8D2D2016 0.27 0.29 -0.28 0.83  
## trt0f2 0.04 0.31 -0.54 0.65  
## trt0f3 0.19 0.31 -0.43 0.79  
## trt0f4 0.07 0.32 -0.53 0.70  
## morph0fB -0.09 0.54 -1.11 1.00  
## morph0fC -1.25 0.34 -1.89 -0.56  
## morph0fD -2.15 0.27 -2.67 -1.62  
## morph0fE -1.11 0.24 -1.59 -0.65  
## morph0fF -1.75 0.74 -3.11 -0.27  
## date0f6D28D2016:trt0f2 0.01 0.46 -0.92 0.89  
## date0f7D12D2016:trt0f2 -0.19 0.39 -0.95 0.60  
## date0f7D18D2016:trt0f2 0.11 0.37 -0.63 0.85  
## date0f7D26D2016:trt0f2 0.15 0.40 -0.61 0.94  
## date0f7D4D2016:trt0f2 0.23 0.42 -0.59 1.05  
## date0f8D2D2016:trt0f2 -0.11 0.41 -0.92 0.70  
## date0f6D28D2016:trt0f3 -0.17 0.54 -1.20 0.91  
## date0f7D12D2016:trt0f3 0.13 0.39 -0.62 0.90  
## date0f7D18D2016:trt0f3 0.19 0.38 -0.55 0.96  
## date0f7D26D2016:trt0f3 0.07 0.40 -0.71 0.86  
## date0f7D4D2016:trt0f3 0.22 0.45 -0.65 1.09  
## date0f8D2D2016:trt0f3 -0.15 0.39 -0.93 0.61  
## date0f6D28D2016:trt0f4 -0.43 0.54 -1.47 0.62  
## date0f7D12D2016:trt0f4 -0.51 0.45 -1.39 0.40  
## date0f7D18D2016:trt0f4 -0.14 0.45 -0.98 0.77  
## date0f7D26D2016:trt0f4 -0.05 0.38 -0.77 0.71  
## date0f7D4D2016:trt0f4 0.35 0.41 -0.44 1.14  
## date0f8D2D2016:trt0f4 0.20 0.39 -0.55 0.98  
## date0f6D28D2016:morph0fB -0.04 0.63 -1.28 1.21  
## date0f7D12D2016:morph0fB -0.01 0.64 -1.25 1.27  
## date0f7D18D2016:morph0fB 0.11 0.83 -1.55 1.70  
## date0f7D26D2016:morph0fB 0.00 0.99 -1.92 1.90  
## date0f7D4D2016:morph0fB -0.00 0.97 -1.93 1.89  
## date0f8D2D2016:morph0fB -0.16 0.58 -1.29 0.98  
## date0f6D28D2016:morph0fC 0.51 0.50 -0.45 1.50  
## date0f7D12D2016:morph0fC 0.54 0.43 -0.29 1.36  
## date0f7D18D2016:morph0fC -0.23 0.42 -1.08 0.58  
## date0f7D26D2016:morph0fC -0.44 0.49 -1.40 0.51  
## date0f7D4D2016:morph0fC -0.28 0.44 -1.14 0.56  
## date0f8D2D2016:morph0fC -0.46 0.44 -1.34 0.40  
## date0f6D28D2016:morph0fD 0.08 0.47 -0.86 1.02  
## date0f7D12D2016:morph0fD 0.20 0.45 -0.65 1.08  
## date0f7D18D2016:morph0fD 0.48 0.41 -0.30 1.29  
## date0f7D26D2016:morph0fD 0.12 0.37 -0.60 0.82  
## date0f7D4D2016:morph0fD 0.10 0.40 -0.66 0.89  
## date0f8D2D2016:morph0fD 0.35 0.38 -0.38 1.09  
## date0f6D28D2016:morph0fE 0.05 0.45 -0.83 0.94  
## date0f7D12D2016:morph0fE -0.40 0.35 -1.09 0.31  
## date0f7D18D2016:morph0fE -0.13 0.36 -0.84 0.60  
## date0f7D26D2016:morph0fE -1.06 0.34 -1.72 -0.39  
## date0f7D4D2016:morph0fE -0.27 0.37 -1.01 0.45  
## date0f8D2D2016:morph0fE -1.13 0.37 -1.85 -0.41  
## date0f6D28D2016:morph0fF -1.18 0.78 -2.69 0.40  
## date0f7D12D2016:morph0fF -0.53 0.86 -2.21 1.14  
## date0f7D18D2016:morph0fF 0.01 1.00 -1.93 1.92  
## date0f7D26D2016:morph0fF 0.01 1.00 -1.99 1.96  
## date0f7D4D2016:morph0fF 0.00 0.99 -1.93 1.91  
## date0f8D2D2016:morph0fF 0.00 0.98 -1.87 1.90  
## trt0f2:morph0fB -0.12 0.60 -1.30 1.09  
## trt0f3:morph0fB 0.21 0.67 -1.15 1.51  
## trt0f4:morph0fB 0.32 0.63 -0.91 1.55  
## trt0f2:morph0fC -0.39 0.43 -1.24 0.45  
## trt0f3:morph0fC -0.58 0.49 -1.53 0.37  
## trt0f4:morph0fC -0.41 0.42 -1.23 0.45  
## trt0f2:morph0fD -0.11 0.38 -0.85 0.63  
## trt0f3:morph0fD -0.10 0.38 -0.86 0.63  
## trt0f4:morph0fD -0.34 0.41 -1.16 0.45  
## trt0f2:morph0fE -0.00 0.33 -0.63 0.67  
## trt0f3:morph0fE 0.04 0.33 -0.60 0.71  
## trt0f4:morph0fE 0.16 0.34 -0.49 0.82  
## trt0f2:morph0fF 0.01 1.00 -1.94 1.93  
## trt0f3:morph0fF -0.00 1.00 -2.01 1.94  
## trt0f4:morph0fF -0.55 0.86 -2.26 1.15  
## date0f6D28D2016:trt0f2:morph0fB 0.08 0.76 -1.38 1.55  
## date0f7D12D2016:trt0f2:morph0fB 0.07 0.75 -1.39 1.54  
## date0f7D18D2016:trt0f2:morph0fB 0.00 1.02 -1.97 2.02  
## date0f7D26D2016:trt0f2:morph0fB -0.01 0.99 -1.96 1.86  
## date0f7D4D2016:trt0f2:morph0fB 0.02 1.01 -1.96 2.09  
## date0f8D2D2016:trt0f2:morph0fB -0.26 0.69 -1.60 1.10  
## date0f6D28D2016:trt0f3:morph0fB 0.14 0.79 -1.40 1.73  
## date0f7D12D2016:trt0f3:morph0fB -0.01 1.01 -2.00 1.99  
## date0f7D18D2016:trt0f3:morph0fB 0.11 0.84 -1.55 1.74  
## date0f7D26D2016:trt0f3:morph0fB -0.01 1.00 -1.93 1.91  
## date0f7D4D2016:trt0f3:morph0fB 0.00 0.98 -1.94 1.92  
## date0f8D2D2016:trt0f3:morph0fB -0.06 0.74 -1.51 1.37  
## date0f6D28D2016:trt0f4:morph0fB 0.37 0.78 -1.17 1.89  
## date0f7D12D2016:trt0f4:morph0fB 0.40 0.73 -1.04 1.82  
## date0f7D18D2016:trt0f4:morph0fB 0.01 1.01 -1.94 2.00  
## date0f7D26D2016:trt0f4:morph0fB 0.00 1.00 -2.03 1.90  
## date0f7D4D2016:trt0f4:morph0fB -0.00 1.02 -1.94 1.99  
## date0f8D2D2016:trt0f4:morph0fB -0.46 0.71 -1.86 0.93  
## date0f6D28D2016:trt0f2:morph0fC -0.46 0.59 -1.58 0.72  
## date0f7D12D2016:trt0f2:morph0fC -0.76 0.56 -1.83 0.34  
## date0f7D18D2016:trt0f2:morph0fC 0.39 0.55 -0.69 1.45  
## date0f7D26D2016:trt0f2:morph0fC 0.17 0.63 -1.04 1.37  
## date0f7D4D2016:trt0f2:morph0fC 0.05 0.57 -1.05 1.20  
## date0f8D2D2016:trt0f2:morph0fC 0.33 0.57 -0.79 1.45  
## date0f6D28D2016:trt0f3:morph0fC -0.20 0.65 -1.51 1.04  
## date0f7D12D2016:trt0f3:morph0fC -1.08 0.59 -2.20 0.07  
## date0f7D18D2016:trt0f3:morph0fC 0.23 0.58 -0.92 1.35  
## date0f7D26D2016:trt0f3:morph0fC 0.01 0.99 -1.91 1.97  
## date0f7D4D2016:trt0f3:morph0fC 0.36 0.61 -0.83 1.54  
## date0f8D2D2016:trt0f3:morph0fC 0.16 0.59 -1.00 1.30  
## date0f6D28D2016:trt0f4:morph0fC -0.14 0.64 -1.42 1.13  
## date0f7D12D2016:trt0f4:morph0fC -0.64 0.59 -1.76 0.49  
## date0f7D18D2016:trt0f4:morph0fC 0.40 0.58 -0.69 1.56  
## date0f7D26D2016:trt0f4:morph0fC 0.26 0.62 -0.97 1.44  
## date0f7D4D2016:trt0f4:morph0fC 0.14 0.55 -0.93 1.23  
## date0f8D2D2016:trt0f4:morph0fC 0.06 0.56 -1.06 1.18  
## date0f6D28D2016:trt0f2:morph0fD -0.18 0.57 -1.27 0.93  
## date0f7D12D2016:trt0f2:morph0fD -0.16 0.56 -1.25 0.94  
## date0f7D18D2016:trt0f2:morph0fD 0.03 0.55 -1.06 1.14  
## date0f7D26D2016:trt0f2:morph0fD -0.20 0.52 -1.22 0.81  
## date0f7D4D2016:trt0f2:morph0fD 0.81 0.53 -0.23 1.87  
## date0f8D2D2016:trt0f2:morph0fD 0.06 0.53 -0.96 1.10  
## date0f6D28D2016:trt0f3:morph0fD 0.20 0.63 -1.02 1.44  
## date0f7D12D2016:trt0f3:morph0fD 0.89 0.59 -0.25 2.03  
## date0f7D18D2016:trt0f3:morph0fD 0.01 0.55 -1.08 1.08  
## date0f7D26D2016:trt0f3:morph0fD 0.16 0.53 -0.88 1.24  
## date0f7D4D2016:trt0f3:morph0fD -0.09 0.56 -1.18 1.01  
## date0f8D2D2016:trt0f3:morph0fD -0.24 0.51 -1.23 0.79  
## date0f6D28D2016:trt0f4:morph0fD -0.57 0.64 -1.81 0.71  
## date0f7D12D2016:trt0f4:morph0fD 0.32 0.62 -0.90 1.55  
## date0f7D18D2016:trt0f4:morph0fD 0.56 0.61 -0.62 1.74  
## date0f7D26D2016:trt0f4:morph0fD 0.03 0.53 -1.05 1.07  
## date0f7D4D2016:trt0f4:morph0fD -0.27 0.58 -1.40 0.89  
## date0f8D2D2016:trt0f4:morph0fD -0.09 0.54 -1.14 0.98  
## date0f6D28D2016:trt0f2:morph0fE -0.30 0.54 -1.37 0.75  
## date0f7D12D2016:trt0f2:morph0fE -0.16 0.49 -1.15 0.80  
## date0f7D18D2016:trt0f2:morph0fE -0.40 0.49 -1.36 0.58  
## date0f7D26D2016:trt0f2:morph0fE -0.02 0.50 -0.99 1.00  
## date0f7D4D2016:trt0f2:morph0fE -0.37 0.51 -1.37 0.66  
## date0f8D2D2016:trt0f2:morph0fE 0.71 0.50 -0.24 1.69  
## date0f6D28D2016:trt0f3:morph0fE -0.28 0.60 -1.46 0.88  
## date0f7D12D2016:trt0f3:morph0fE -0.06 0.49 -1.04 0.89  
## date0f7D18D2016:trt0f3:morph0fE -0.14 0.52 -1.17 0.84  
## date0f7D26D2016:trt0f3:morph0fE -0.03 0.51 -1.03 0.98  
## date0f7D4D2016:trt0f3:morph0fE -0.30 0.52 -1.30 0.70  
## date0f8D2D2016:trt0f3:morph0fE 0.05 0.51 -0.96 1.03  
## date0f6D28D2016:trt0f4:morph0fE -0.05 0.59 -1.16 1.12  
## date0f7D12D2016:trt0f4:morph0fE 0.03 0.55 -1.05 1.12  
## date0f7D18D2016:trt0f4:morph0fE -0.28 0.56 -1.37 0.82  
## date0f7D26D2016:trt0f4:morph0fE -0.28 0.48 -1.18 0.67  
## date0f7D4D2016:trt0f4:morph0fE -0.63 0.50 -1.63 0.38  
## date0f8D2D2016:trt0f4:morph0fE -0.13 0.49 -1.06 0.83  
## date0f6D28D2016:trt0f2:morph0fF -0.01 0.99 -1.94 1.91  
## date0f7D12D2016:trt0f2:morph0fF 0.02 0.99 -1.93 1.96  
## date0f7D18D2016:trt0f2:morph0fF -0.01 0.97 -1.91 1.82  
## date0f7D26D2016:trt0f2:morph0fF -0.01 0.99 -2.00 1.92  
## date0f7D4D2016:trt0f2:morph0fF 0.02 1.03 -1.98 2.04  
## date0f8D2D2016:trt0f2:morph0fF -0.01 1.00 -1.94 1.96  
## date0f6D28D2016:trt0f3:morph0fF -0.01 0.97 -1.87 1.94  
## date0f7D12D2016:trt0f3:morph0fF 0.03 1.01 -1.98 2.02  
## date0f7D18D2016:trt0f3:morph0fF -0.00 1.03 -1.98 2.03  
## date0f7D26D2016:trt0f3:morph0fF -0.02 0.97 -1.95 1.91  
## date0f7D4D2016:trt0f3:morph0fF -0.01 1.02 -1.97 1.90  
## date0f8D2D2016:trt0f3:morph0fF 0.00 0.98 -1.83 1.93  
## date0f6D28D2016:trt0f4:morph0fF -0.04 1.02 -2.06 2.00  
## date0f7D12D2016:trt0f4:morph0fF -0.56 0.86 -2.24 1.16  
## date0f7D18D2016:trt0f4:morph0fF 0.00 1.01 -2.07 1.95  
## date0f7D26D2016:trt0f4:morph0fF -0.02 1.02 -2.01 1.92  
## date0f7D4D2016:trt0f4:morph0fF 0.00 1.00 -1.92 1.96  
## date0f8D2D2016:trt0f4:morph0fF 0.01 0.99 -1.94 1.94  
## Eff.Sample Rhat  
## Intercept 3222 1.00  
## date0f6D28D2016 2804 1.00  
## date0f7D12D2016 3320 1.00  
## date0f7D18D2016 2964 1.00  
## date0f7D26D2016 3378 1.00  
## date0f7D4D2016 3569 1.00  
## date0f8D2D2016 3208 1.00  
## trt0f2 3472 1.00  
## trt0f3 3199 1.00  
## trt0f4 3440 1.00  
## morph0fB 4000 1.00  
## morph0fC 3375 1.00  
## morph0fD 3552 1.00  
## morph0fE 3245 1.00  
## morph0fF 4000 1.00  
## date0f6D28D2016:trt0f2 3519 1.00  
## date0f7D12D2016:trt0f2 4000 1.00  
## date0f7D18D2016:trt0f2 3434 1.00  
## date0f7D26D2016:trt0f2 4000 1.00  
## date0f7D4D2016:trt0f2 4000 1.00  
## date0f8D2D2016:trt0f2 4000 1.00  
## date0f6D28D2016:trt0f3 4000 1.00  
## date0f7D12D2016:trt0f3 4000 1.00  
## date0f7D18D2016:trt0f3 3233 1.00  
## date0f7D26D2016:trt0f3 4000 1.00  
## date0f7D4D2016:trt0f3 4000 1.00  
## date0f8D2D2016:trt0f3 3603 1.00  
## date0f6D28D2016:trt0f4 4000 1.00  
## date0f7D12D2016:trt0f4 4000 1.00  
## date0f7D18D2016:trt0f4 4000 1.00  
## date0f7D26D2016:trt0f4 4000 1.00  
## date0f7D4D2016:trt0f4 4000 1.00  
## date0f8D2D2016:trt0f4 3718 1.00  
## date0f6D28D2016:morph0fB 4000 1.00  
## date0f7D12D2016:morph0fB 4000 1.00  
## date0f7D18D2016:morph0fB 4000 1.00  
## date0f7D26D2016:morph0fB 4000 1.00  
## date0f7D4D2016:morph0fB 4000 1.00  
## date0f8D2D2016:morph0fB 4000 1.00  
## date0f6D28D2016:morph0fC 3328 1.00  
## date0f7D12D2016:morph0fC 3751 1.00  
## date0f7D18D2016:morph0fC 3743 1.00  
## date0f7D26D2016:morph0fC 4000 1.00  
## date0f7D4D2016:morph0fC 4000 1.00  
## date0f8D2D2016:morph0fC 3607 1.00  
## date0f6D28D2016:morph0fD 4000 1.00  
## date0f7D12D2016:morph0fD 4000 1.00  
## date0f7D18D2016:morph0fD 4000 1.00  
## date0f7D26D2016:morph0fD 4000 1.00  
## date0f7D4D2016:morph0fD 3747 1.00  
## date0f8D2D2016:morph0fD 3523 1.00  
## date0f6D28D2016:morph0fE 3146 1.00  
## date0f7D12D2016:morph0fE 4000 1.00  
## date0f7D18D2016:morph0fE 4000 1.00  
## date0f7D26D2016:morph0fE 4000 1.00  
## date0f7D4D2016:morph0fE 3746 1.00  
## date0f8D2D2016:morph0fE 3444 1.00  
## date0f6D28D2016:morph0fF 4000 1.00  
## date0f7D12D2016:morph0fF 4000 1.00  
## date0f7D18D2016:morph0fF 4000 1.00  
## date0f7D26D2016:morph0fF 4000 1.00  
## date0f7D4D2016:morph0fF 4000 1.00  
## date0f8D2D2016:morph0fF 4000 1.00  
## trt0f2:morph0fB 4000 1.00  
## trt0f3:morph0fB 4000 1.00  
## trt0f4:morph0fB 4000 1.00  
## trt0f2:morph0fC 4000 1.00  
## trt0f3:morph0fC 3568 1.00  
## trt0f4:morph0fC 4000 1.00  
## trt0f2:morph0fD 4000 1.00  
## trt0f3:morph0fD 3633 1.00  
## trt0f4:morph0fD 4000 1.00  
## trt0f2:morph0fE 3501 1.00  
## trt0f3:morph0fE 3492 1.00  
## trt0f4:morph0fE 3480 1.00  
## trt0f2:morph0fF 4000 1.00  
## trt0f3:morph0fF 4000 1.00  
## trt0f4:morph0fF 4000 1.00  
## date0f6D28D2016:trt0f2:morph0fB 4000 1.00  
## date0f7D12D2016:trt0f2:morph0fB 4000 1.00  
## date0f7D18D2016:trt0f2:morph0fB 4000 1.00  
## date0f7D26D2016:trt0f2:morph0fB 4000 1.00  
## date0f7D4D2016:trt0f2:morph0fB 4000 1.00  
## date0f8D2D2016:trt0f2:morph0fB 4000 1.00  
## date0f6D28D2016:trt0f3:morph0fB 4000 1.00  
## date0f7D12D2016:trt0f3:morph0fB 4000 1.00  
## date0f7D18D2016:trt0f3:morph0fB 4000 1.00  
## date0f7D26D2016:trt0f3:morph0fB 4000 1.00  
## date0f7D4D2016:trt0f3:morph0fB 4000 1.00  
## date0f8D2D2016:trt0f3:morph0fB 4000 1.00  
## date0f6D28D2016:trt0f4:morph0fB 4000 1.00  
## date0f7D12D2016:trt0f4:morph0fB 4000 1.00  
## date0f7D18D2016:trt0f4:morph0fB 4000 1.00  
## date0f7D26D2016:trt0f4:morph0fB 4000 1.00  
## date0f7D4D2016:trt0f4:morph0fB 4000 1.00  
## date0f8D2D2016:trt0f4:morph0fB 4000 1.00  
## date0f6D28D2016:trt0f2:morph0fC 4000 1.00  
## date0f7D12D2016:trt0f2:morph0fC 4000 1.00  
## date0f7D18D2016:trt0f2:morph0fC 4000 1.00  
## date0f7D26D2016:trt0f2:morph0fC 4000 1.00  
## date0f7D4D2016:trt0f2:morph0fC 4000 1.00  
## date0f8D2D2016:trt0f2:morph0fC 4000 1.00  
## date0f6D28D2016:trt0f3:morph0fC 4000 1.00  
## date0f7D12D2016:trt0f3:morph0fC 4000 1.00  
## date0f7D18D2016:trt0f3:morph0fC 4000 1.00  
## date0f7D26D2016:trt0f3:morph0fC 4000 1.00  
## date0f7D4D2016:trt0f3:morph0fC 4000 1.00  
## date0f8D2D2016:trt0f3:morph0fC 4000 1.00  
## date0f6D28D2016:trt0f4:morph0fC 4000 1.00  
## date0f7D12D2016:trt0f4:morph0fC 4000 1.00  
## date0f7D18D2016:trt0f4:morph0fC 4000 1.00  
## date0f7D26D2016:trt0f4:morph0fC 4000 1.00  
## date0f7D4D2016:trt0f4:morph0fC 4000 1.00  
## date0f8D2D2016:trt0f4:morph0fC 4000 1.00  
## date0f6D28D2016:trt0f2:morph0fD 4000 1.00  
## date0f7D12D2016:trt0f2:morph0fD 4000 1.00  
## date0f7D18D2016:trt0f2:morph0fD 4000 1.00  
## date0f7D26D2016:trt0f2:morph0fD 4000 1.00  
## date0f7D4D2016:trt0f2:morph0fD 4000 1.00  
## date0f8D2D2016:trt0f2:morph0fD 4000 1.00  
## date0f6D28D2016:trt0f3:morph0fD 4000 1.00  
## date0f7D12D2016:trt0f3:morph0fD 4000 1.00  
## date0f7D18D2016:trt0f3:morph0fD 4000 1.00  
## date0f7D26D2016:trt0f3:morph0fD 4000 1.00  
## date0f7D4D2016:trt0f3:morph0fD 4000 1.00  
## date0f8D2D2016:trt0f3:morph0fD 4000 1.00  
## date0f6D28D2016:trt0f4:morph0fD 4000 1.00  
## date0f7D12D2016:trt0f4:morph0fD 4000 1.00  
## date0f7D18D2016:trt0f4:morph0fD 4000 1.00  
## date0f7D26D2016:trt0f4:morph0fD 4000 1.00  
## date0f7D4D2016:trt0f4:morph0fD 4000 1.00  
## date0f8D2D2016:trt0f4:morph0fD 4000 1.00  
## date0f6D28D2016:trt0f2:morph0fE 4000 1.00  
## date0f7D12D2016:trt0f2:morph0fE 4000 1.00  
## date0f7D18D2016:trt0f2:morph0fE 4000 1.00  
## date0f7D26D2016:trt0f2:morph0fE 4000 1.00  
## date0f7D4D2016:trt0f2:morph0fE 4000 1.00  
## date0f8D2D2016:trt0f2:morph0fE 4000 1.00  
## date0f6D28D2016:trt0f3:morph0fE 4000 1.00  
## date0f7D12D2016:trt0f3:morph0fE 4000 1.00  
## date0f7D18D2016:trt0f3:morph0fE 4000 1.00  
## date0f7D26D2016:trt0f3:morph0fE 4000 1.00  
## date0f7D4D2016:trt0f3:morph0fE 4000 1.00  
## date0f8D2D2016:trt0f3:morph0fE 4000 1.00  
## date0f6D28D2016:trt0f4:morph0fE 4000 1.00  
## date0f7D12D2016:trt0f4:morph0fE 4000 1.00  
## date0f7D18D2016:trt0f4:morph0fE 4000 1.00  
## date0f7D26D2016:trt0f4:morph0fE 4000 1.00  
## date0f7D4D2016:trt0f4:morph0fE 4000 1.00  
## date0f8D2D2016:trt0f4:morph0fE 4000 1.00  
## date0f6D28D2016:trt0f2:morph0fF 4000 1.00  
## date0f7D12D2016:trt0f2:morph0fF 4000 1.00  
## date0f7D18D2016:trt0f2:morph0fF 4000 1.00  
## date0f7D26D2016:trt0f2:morph0fF 4000 1.00  
## date0f7D4D2016:trt0f2:morph0fF 4000 1.00  
## date0f8D2D2016:trt0f2:morph0fF 4000 1.00  
## date0f6D28D2016:trt0f3:morph0fF 4000 1.00  
## date0f7D12D2016:trt0f3:morph0fF 4000 1.00  
## date0f7D18D2016:trt0f3:morph0fF 4000 1.00  
## date0f7D26D2016:trt0f3:morph0fF 4000 1.00  
## date0f7D4D2016:trt0f3:morph0fF 4000 1.00  
## date0f8D2D2016:trt0f3:morph0fF 4000 1.00  
## date0f6D28D2016:trt0f4:morph0fF 4000 1.00  
## date0f7D12D2016:trt0f4:morph0fF 4000 1.00  
## date0f7D18D2016:trt0f4:morph0fF 4000 1.00  
## date0f7D26D2016:trt0f4:morph0fF 4000 1.00  
## date0f7D4D2016:trt0f4:morph0fF 4000 1.00  
## date0f8D2D2016:trt0f4:morph0fF 4000 1.00  
##   
## Family Specific Parameters:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## shape 2.51 0.18 2.17 2.87 4000 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(chirobody,type="hist")

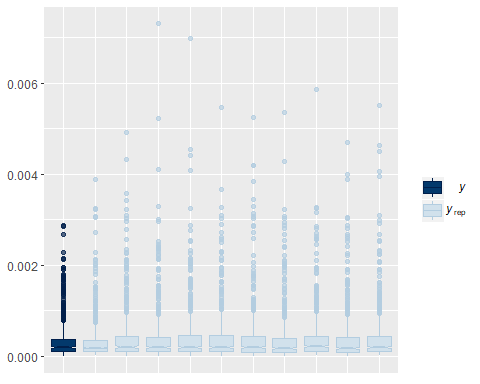
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

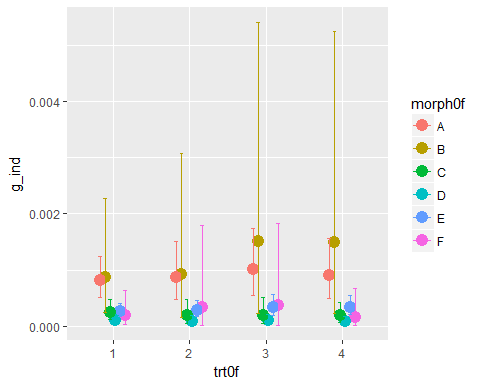
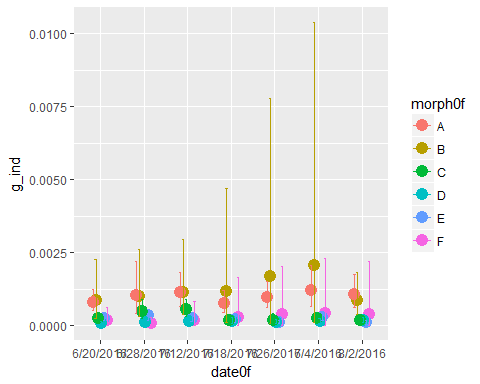
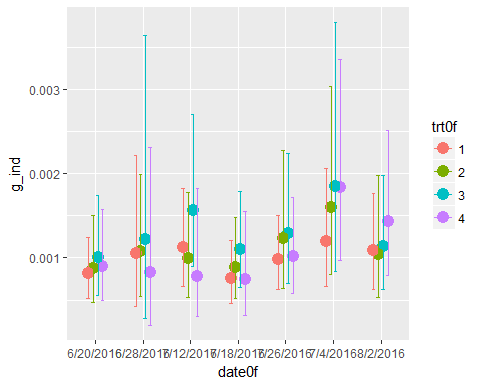
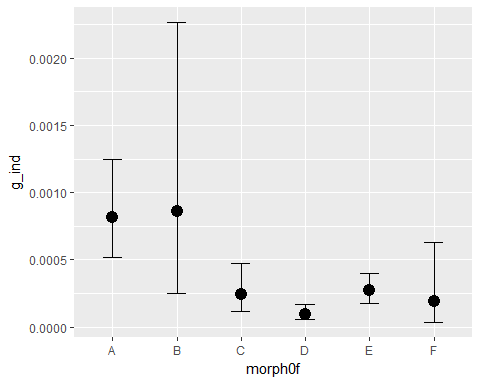
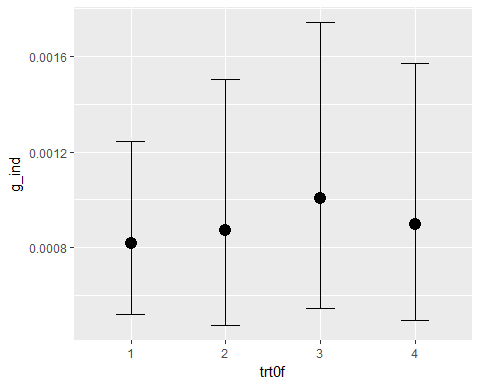
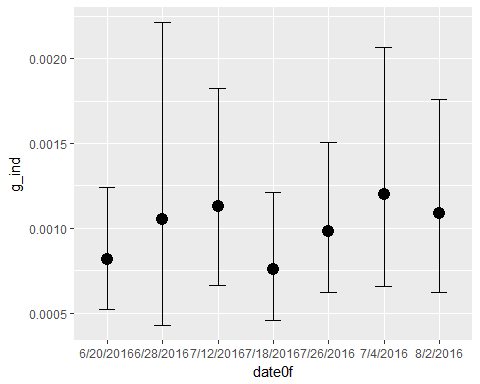


pp\_check(chirobody,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



marginal\_effects(chirobody, robust=FALSE)



Effect of Treatment on Chironomid Body Size

get\_prior(g\_ind~date0f\*trt0f+(1|tank),data=dragon\_g\_ind,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D26D2016   
## 3 b date0f7D26D2016:trt0f2   
## 4 b date0f7D26D2016:trt0f3   
## 5 b date0f7D26D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, -3, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## 17 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17

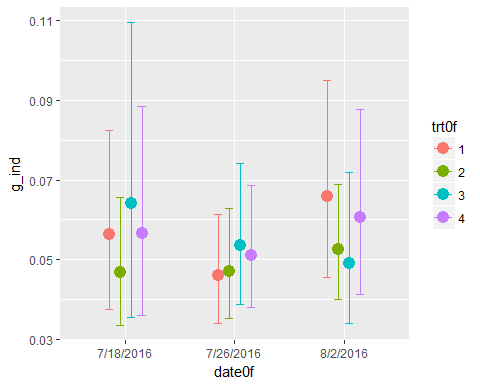
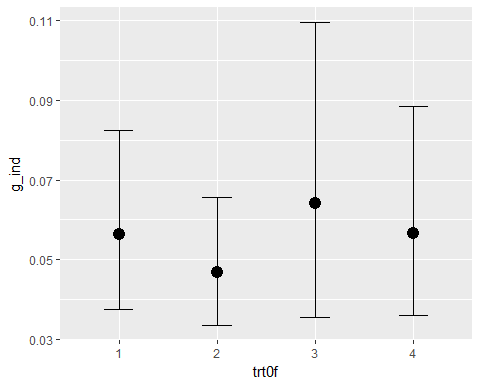
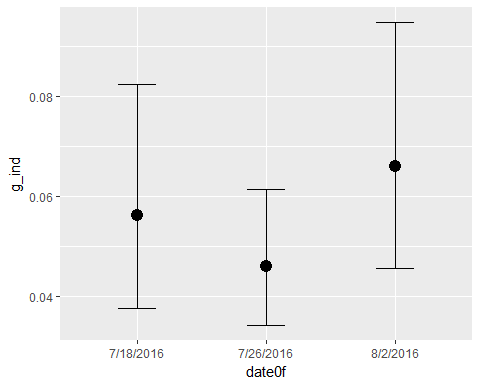
dragonbody<-brm(g\_ind~date0f\*trt0f+(1|tank),data=dragon\_g\_ind,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,1),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.921 seconds (Warm-up)  
## 1.211 seconds (Sampling)  
## 3.132 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.819 seconds (Warm-up)  
## 0.952 seconds (Sampling)  
## 2.771 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.668 seconds (Warm-up)  
## 1.551 seconds (Sampling)  
## 3.219 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 1.868 seconds (Warm-up)  
## 1.66 seconds (Sampling)  
## 3.528 seconds (Total)

marginal\_effects(dragonbody, robust=FALSE)



Effect of Treatment on Chironomid Body Size

get\_prior(g\_ind~date0f\*trt0f+(1|tank),data=may\_g\_ind,family=Gamma(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f04MJulM16   
## 3 b date0f04MJulM16:trt0f2   
## 4 b date0f04MJulM16:trt0f3   
## 5 b date0f04MJulM16:trt0f4   
## 6 b date0f12MJulM16   
## 7 b date0f12MJulM16:trt0f2   
## 8 b date0f12MJulM16:trt0f3   
## 9 b date0f12MJulM16:trt0f4   
## 10 b date0f26MJulM16   
## 11 b date0f26MJulM16:trt0f2   
## 12 b date0f26MJulM16:trt0f3   
## 13 b date0f26MJulM16:trt0f4   
## 14 b date0f28MJunM16   
## 15 b date0f28MJunM16:trt0f2   
## 16 b date0f28MJunM16:trt0f3   
## 17 b date0f28MJunM16:trt0f4   
## 18 b trt0f2   
## 19 b trt0f3   
## 20 b trt0f4   
## 21 student\_t(3, -6, 10) Intercept   
## 22 student\_t(3, 0, 10) sd   
## 23 sd tank   
## 24 sd Intercept tank   
## 25 gamma(0.01, 0.01) shape   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16   
## 17   
## 18   
## 19   
## 20   
## 21   
## 22   
## 23   
## 24   
## 25

maybody<-brm(g\_ind~date0f\*trt0f+(1|tank),data=may\_g\_ind,family=Gamma(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,1),class=b)))

## Compiling the C++ model

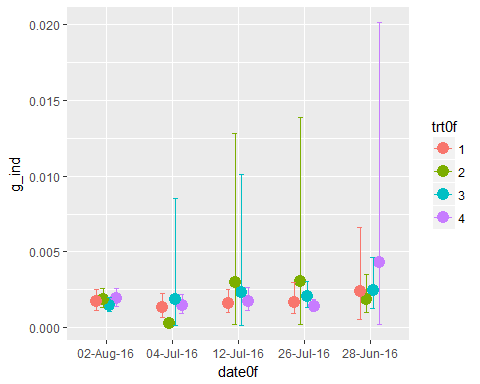
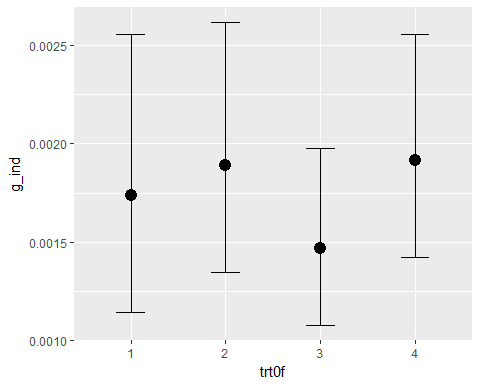
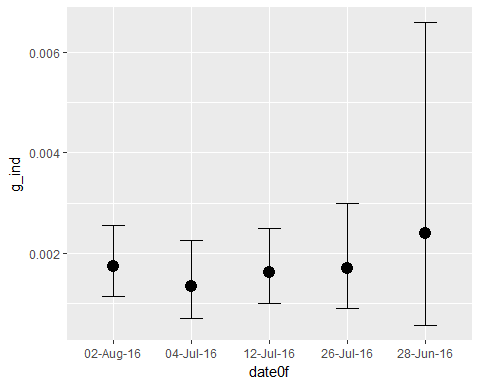
## Start sampling

##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.429 seconds (Warm-up)  
## 1.787 seconds (Sampling)  
## 4.216 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.101 seconds (Warm-up)  
## 1.793 seconds (Sampling)  
## 3.894 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.44 seconds (Warm-up)  
## 1.659 seconds (Sampling)  
## 4.099 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 2.278 seconds (Warm-up)  
## 1.849 seconds (Sampling)  
## 4.127 seconds (Total)

## Warning: There were 1 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help. See  
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

marginal\_effects(maybody, robust=FALSE)



Effects of Dragonflies in the Last 3 Weeks

Data

urlfile8<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/odononew.csv"  
odononew<-read.csv(urlfile8)  
  
odononew$trt0f<-as.factor(odononew$trt)  
odononew$date0f<-as.factor(odononew$date)

Model with just Date and Treatment as Predictors for last 3 Weeks

get\_prior(totnonew~date0f\*trt0f +(1|tank),data=odononew,family=poisson(link="log"))

## prior class coef group resp dpar  
## 1 b   
## 2 b date0f7D26D2016   
## 3 b date0f7D26D2016:trt0f2   
## 4 b date0f7D26D2016:trt0f3   
## 5 b date0f7D26D2016:trt0f4   
## 6 b date0f8D2D2016   
## 7 b date0f8D2D2016:trt0f2   
## 8 b date0f8D2D2016:trt0f3   
## 9 b date0f8D2D2016:trt0f4   
## 10 b trt0f2   
## 11 b trt0f3   
## 12 b trt0f4   
## 13 student\_t(3, 3, 10) Intercept   
## 14 student\_t(3, 0, 10) sd   
## 15 sd tank   
## 16 sd Intercept tank   
## nlpar bound  
## 1   
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9   
## 10   
## 11   
## 12   
## 13   
## 14   
## 15   
## 16

abundadultend<-brm(totno~date0f\*trt0f +(1|tank),data=odononew,family=poisson(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,5),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.071 seconds (Warm-up)  
## 6.356 seconds (Sampling)  
## 12.427 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 6.335 seconds (Warm-up)  
## 7.044 seconds (Sampling)  
## 13.379 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.343 seconds (Warm-up)  
## 6.234 seconds (Sampling)  
## 11.577 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0.001 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 5.962 seconds (Warm-up)  
## 6.331 seconds (Sampling)  
## 12.293 seconds (Total)

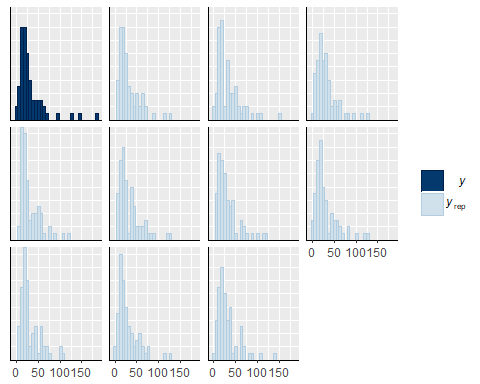
print(abundadultend)

## Family: poisson   
## Links: mu = log   
## Formula: totno ~ date0f \* trt0f + (1 | tank)   
## Data: odononew (Number of observations: 70)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.52 0.09 0.38 0.74 1434 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample  
## Intercept 2.89 0.24 2.41 3.35 1611  
## date0f7D26D2016 0.36 0.12 0.13 0.60 2853  
## date0f8D2D2016 1.24 0.10 1.04 1.45 2621  
## trt0f2 0.07 0.33 -0.54 0.72 1733  
## trt0f3 0.27 0.33 -0.37 0.93 1913  
## trt0f4 -0.40 0.34 -1.07 0.26 1920  
## date0f7D26D2016:trt0f2 -0.83 0.18 -1.18 -0.47 3183  
## date0f8D2D2016:trt0f2 -0.50 0.15 -0.78 -0.21 3006  
## date0f7D26D2016:trt0f3 -0.26 0.17 -0.59 0.06 3040  
## date0f8D2D2016:trt0f3 -0.52 0.14 -0.79 -0.24 3048  
## date0f7D26D2016:trt0f4 -0.24 0.20 -0.64 0.16 3194  
## date0f8D2D2016:trt0f4 -0.18 0.18 -0.52 0.16 3071  
## Rhat  
## Intercept 1.01  
## date0f7D26D2016 1.00  
## date0f8D2D2016 1.00  
## trt0f2 1.00  
## trt0f3 1.00  
## trt0f4 1.00  
## date0f7D26D2016:trt0f2 1.00  
## date0f8D2D2016:trt0f2 1.00  
## date0f7D26D2016:trt0f3 1.00  
## date0f8D2D2016:trt0f3 1.00  
## date0f7D26D2016:trt0f4 1.00  
## date0f8D2D2016:trt0f4 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(abundadultend,type="hist")

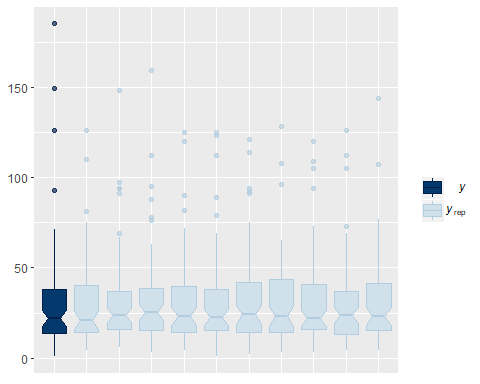
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(abundadultend,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



Model with Treatment and Dragonfly Abundance as Predictors, and Date as a Random Effect for last 3 Weeks

odomodel2<-brm(totno~trt0f\*no\_odo +(1|tank/date0f),data=odononew,family=poisson(link="log"),  
 prior=c(prior(cauchy(0,1),class=sd),  
 prior(normal(0,5),class=b)))

## Compiling the C++ model

## Start sampling

##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 1).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 18.569 seconds (Warm-up)  
## 12.414 seconds (Sampling)  
## 30.983 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 2).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 20.682 seconds (Warm-up)  
## 12.282 seconds (Sampling)  
## 32.964 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 3).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 14.324 seconds (Warm-up)  
## 12.166 seconds (Sampling)  
## 26.49 seconds (Total)  
##   
##   
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 4).  
##   
## Gradient evaluation took 0 seconds  
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Adjust your expectations accordingly!  
##   
##   
## Iteration: 1 / 2000 [ 0%] (Warmup)  
## Iteration: 200 / 2000 [ 10%] (Warmup)  
## Iteration: 400 / 2000 [ 20%] (Warmup)  
## Iteration: 600 / 2000 [ 30%] (Warmup)  
## Iteration: 800 / 2000 [ 40%] (Warmup)  
## Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Iteration: 2000 / 2000 [100%] (Sampling)  
##   
## Elapsed Time: 14.358 seconds (Warm-up)  
## 12.132 seconds (Sampling)  
## 26.49 seconds (Total)

## Warning: There were 14 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help. See  
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

print(odomodel2)

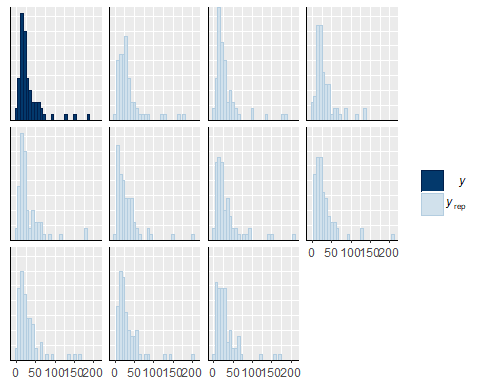
## Warning: There were 14 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help.  
## See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Family: poisson   
## Links: mu = log   
## Formula: totno ~ trt0f \* no\_odo + (1 | tank/date0f)   
## Data: odononew (Number of observations: 70)   
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;   
## total post-warmup samples = 4000  
## ICs: LOO = NA; WAIC = NA; R2 = NA  
##   
## Group-Level Effects:   
## ~tank (Number of levels: 24)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.17 0.13 0.01 0.46 621 1.00  
## sd:date0f(Intercept) 0.81 0.09 0.67 1.00 1588 1.00  
##   
## ~tank:date0f (Number of levels: 70)   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## sd(Intercept) 0.81 0.09 0.67 1.00 1588 1.00  
##   
## Population-Level Effects:   
## Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat  
## Intercept 3.14 0.26 2.64 3.64 1199 1.00  
## trt0f2 -0.18 0.41 -1.01 0.64 1750 1.00  
## trt0f3 0.17 0.37 -0.57 0.89 1557 1.00  
## trt0f4 -0.49 0.41 -1.29 0.35 1435 1.00  
## no\_odo 0.14 0.13 -0.10 0.40 1471 1.00  
## trt0f2:no\_odo -0.12 0.19 -0.48 0.25 1776 1.00  
## trt0f3:no\_odo -0.13 0.15 -0.43 0.17 1661 1.00  
## trt0f4:no\_odo -0.05 0.17 -0.39 0.28 1620 1.00  
##   
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample   
## is a crude measure of effective sample size, and Rhat is the potential   
## scale reduction factor on split chains (at convergence, Rhat = 1).

pp\_check(odomodel2,type="hist")

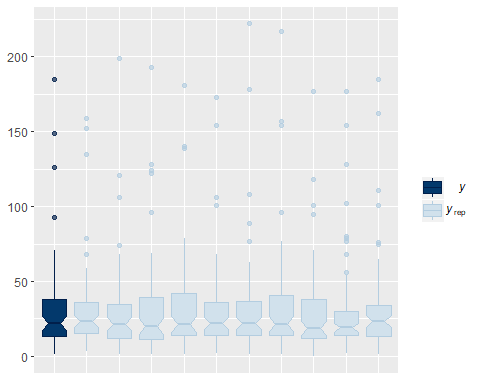
## Using 10 posterior samples for ppc type 'hist' by default.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



pp\_check(odomodel2,type="boxplot")

## Using 10 posterior samples for ppc type 'boxplot' by default.



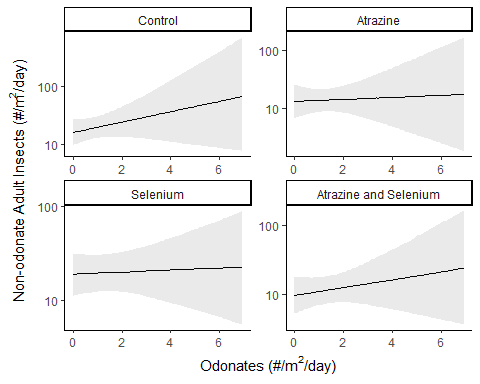
Model Comparison

waic(odomodel2, abundadultend)

## WAIC SE  
## odomodel2 464.79 6.98  
## abundadultend 1151.36 134.52  
## odomodel2 - abundadultend -686.57 132.79

Odonate Model Plot

odomplot<-marginal\_effects(odomodel2)  
odomplot<-data.frame(odomplot$`no\_odo:trt0f`)  
odomplot$trt2<-ifelse(odomplot$trt0f=="1", "Control",  
 ifelse(odomplot$trt0f=="2", "Atrazine",  
 ifelse(odomplot$trt0f=="3", "Selenium", "Atrazine and Selenium")))  
odomplot$trt3<-factor(odomplot$trt2, levels=c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"))  
odomplot$est<-(odomplot$estimate\_\_)\*2.78/4  
odomplot$lower<-(odomplot$lower\_\_)\*2.78/4  
odomplot$upper<-(odomplot$upper\_\_)\*2.78/4  
odomplot$odo<-(odomplot$no\_odo)\*2.78/4  
  
odomodelplot<-ggplot(odomplot, aes(x=odo, y=est, ymin=lower, ymax=upper, group=trt3))+  
 geom\_line()+  
 geom\_ribbon(alpha=0.1)+  
 ylab(expression(paste("Non-odonate Adult Insects (#/m"^2,"/day)")))+  
 xlab(expression(paste("Odonates (#/m"^2,"/day)")))+  
 theme\_classic()+  
 scale\_y\_log10()+  
 facet\_wrap(~trt3, scales="free")  
  
odomodelplot



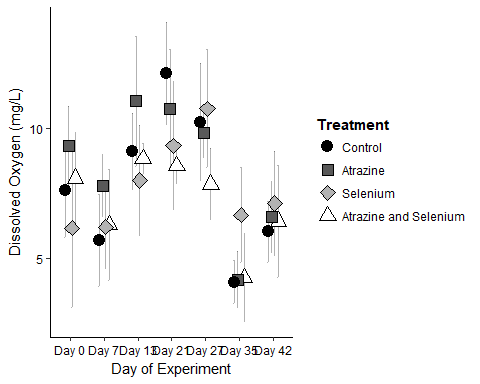
Water Quality Data Plots

Data

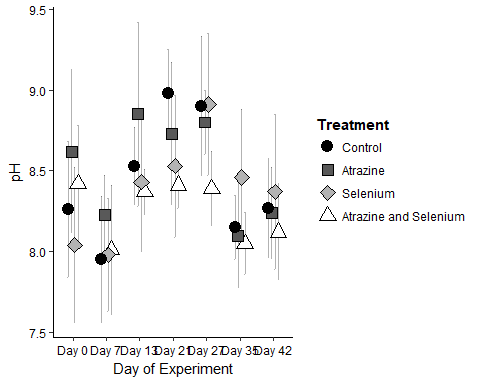
urlfile9<-"https://raw.githubusercontent.com/briannahenry/Mesocosm2/master/WQ\_new.csv"  
WQ<-read.csv(urlfile9)  
  
WQ$date2<-factor(WQ$date, levels = c("6/20/2016","6/28/2016", "7/4/2016", "7/12/2016", "7/18/2016", "7/26/2016", "8/2/2016"))  
WQ$trt0f<-as.factor(WQ$trt)  
  
WQ$date3<-ifelse(WQ$date2=="6/20/2016", "Day 0",  
 ifelse(WQ$date2=="6/28/2016", "Day 7",  
 ifelse(WQ$date2=="7/4/2016", "Day 13",  
 ifelse(WQ$date2=="7/12/2016", "Day 21",  
 ifelse(WQ$date2=="7/18/2016", "Day 27",  
 ifelse(WQ$date2=="7/26/2016", "Day 35", "Day 42"))))))  
WQ$date4<-factor(WQ$date3, levels=c("Day 0", "Day 7", "Day 13", "Day 21", "Day 27", "Day 35", "Day 42"))

Plots

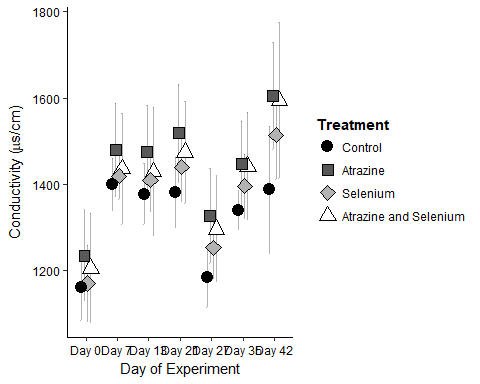
######Dissolved Oxygen#####  
DOplot<-ggplot(WQ,aes(x=date4,y=do,ymin=sdlow2,ymax=sdhi2,fill=trt0f, shape=trt0f))+  
 geom\_errorbar(position=position\_dodge(width=0.4), alpha=0.3, width=0.1)+  
 geom\_point(size=4,position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab("Dissolved Oxygen (mg/L)")+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))   
  
DOplot



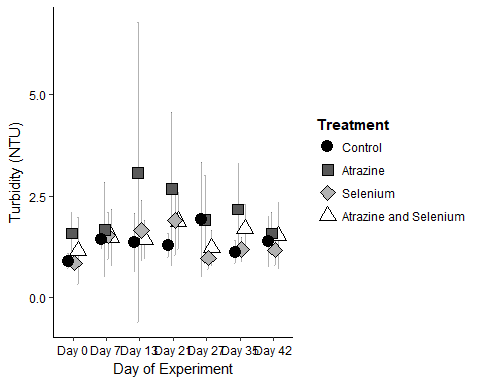
#####PH####  
PHplot<-ggplot(WQ,aes(x=date4,y=ph,ymin=sdlow,ymax=sdhi,fill=trt0f, shape=trt0f))+  
 geom\_errorbar(position=position\_dodge(width=0.4), alpha=0.3, width=0.1)+  
 geom\_point(size=4,position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab("pH")+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))   
  
PHplot



#####Conductivity#####  
condplot<-ggplot(WQ,aes(x=date4,y=cond,ymin=sdlow3,ymax=sdhi3,fill=trt0f, shape=trt0f))+  
 geom\_errorbar(position=position\_dodge(width=0.4), alpha=0.3, width=0.1)+  
 geom\_point(size=4,position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab(expression(paste("Conductivity (", mu, "s/cm)")))+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))   
  
condplot



####Turbidity#####  
turbplot<-ggplot(WQ,aes(x=date4,y=turb,ymin=sdlow4,ymax=sdhi4,fill=trt0f, shape=trt0f))+  
 geom\_errorbar(position=position\_dodge(width=0.4), alpha=0.3, width=0.1)+  
 geom\_point(size=4,position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab("Turbidity (NTU)")+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))   
  
turbplot



####Temeprature####  
tempplot<-ggplot(WQ,aes(x=date4,y=temp,ymin=sdlow5,ymax=sdhi5,fill=trt0f, shape=trt0f))+  
 geom\_errorbar(position=position\_dodge(width=0.4), alpha=0.3, width=0.1)+  
 geom\_point(size=4,position=position\_dodge(width=0.4))+  
 xlab("Day of Experiment")+  
 ylab(expression("Temperature " ( degree\*C)))+  
 scale\_shape\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c(21,22,23,24))+  
 scale\_fill\_manual(name="Treatment", labels = c("Control", "Atrazine", "Selenium", "Atrazine and Selenium"), values = c("grey0","grey35","grey70","grey100"))+  
 theme\_classic()+  
 theme(axis.text.x = element\_text(color="black"))+  
 theme(axis.text.y = element\_text(color="black"))+  
 theme(legend.title = element\_text(face="bold", size="11"))   
  
tempplot

