Assignment7\_B04704016林家毅

# Question 1

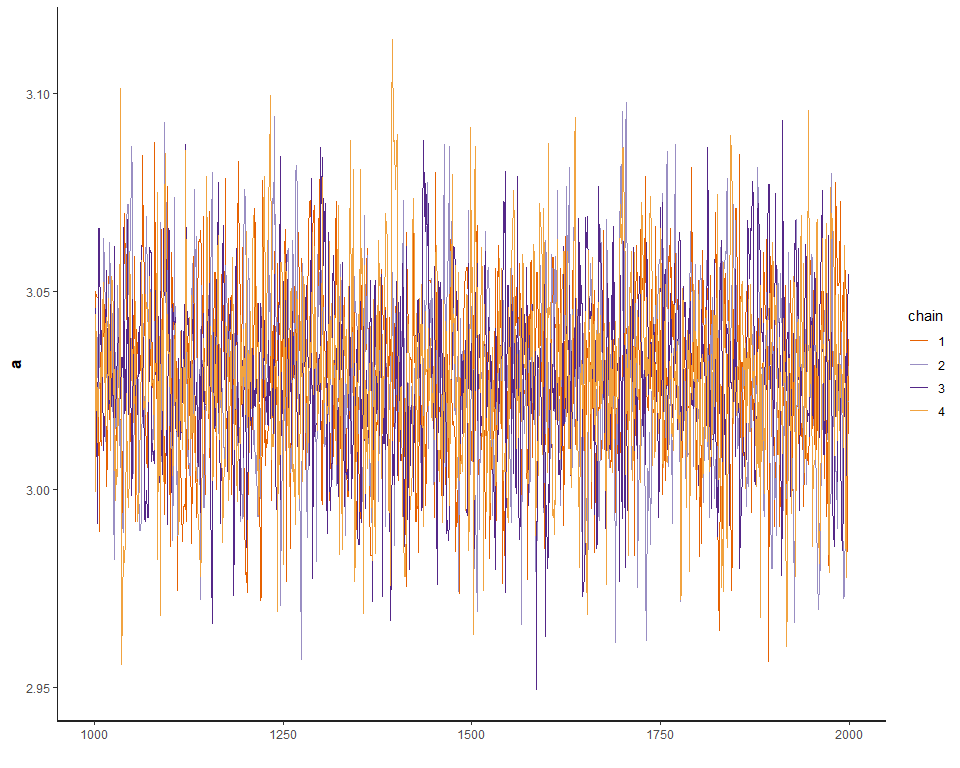
# Load data  
data( "Hurricanes" )  
data <- Hurricanes; rm( Hurricanes )  
str( data )

## 'data.frame': 92 obs. of 8 variables:  
## $ name : Factor w/ 83 levels "Able","Agnes",..: 38 77 1 9 47 20 40 60 27 33 ...  
## $ year : int 1950 1950 1952 1953 1953 1954 1954 1954 1955 1955 ...  
## $ deaths : int 2 4 3 1 0 60 20 20 0 200 ...  
## $ category : int 3 3 1 1 1 3 3 4 3 1 ...  
## $ min\_pressure: int 960 955 985 987 985 960 954 938 962 987 ...  
## $ damage\_norm : int 1590 5350 150 58 15 19321 3230 24260 2030 14730 ...  
## $ female : int 1 0 0 1 1 1 1 1 1 1 ...  
## $ femininity : num 6.78 1.39 3.83 9.83 8.33 ...

# Create data list for the stan model  
model.data <- list(  
 N = NROW( data ),  
 D = data$deaths,  
 C = data$category,  
 MP = scale( data$min\_pressure )[ ,1],  
 DN = scale( data$damage\_norm )[ ,1],  
 FE = data$female,  
 FY = data$femininity  
)  
str( model.data )

## List of 7  
## $ N : int 92  
## $ D : int [1:92] 2 4 3 1 0 60 20 20 0 200 ...  
## $ C : int [1:92] 3 3 1 1 1 3 3 4 3 1 ...  
## $ MP: num [1:92] -0.258 -0.52 1.054 1.158 1.054 ...  
## $ DN: num [1:92] -0.439 -0.148 -0.55 -0.558 -0.561 ...  
## $ FE: int [1:92] 1 0 0 1 1 1 1 1 1 1 ...  
## $ FY: num [1:92] 6.78 1.39 3.83 9.83 8.33 ...

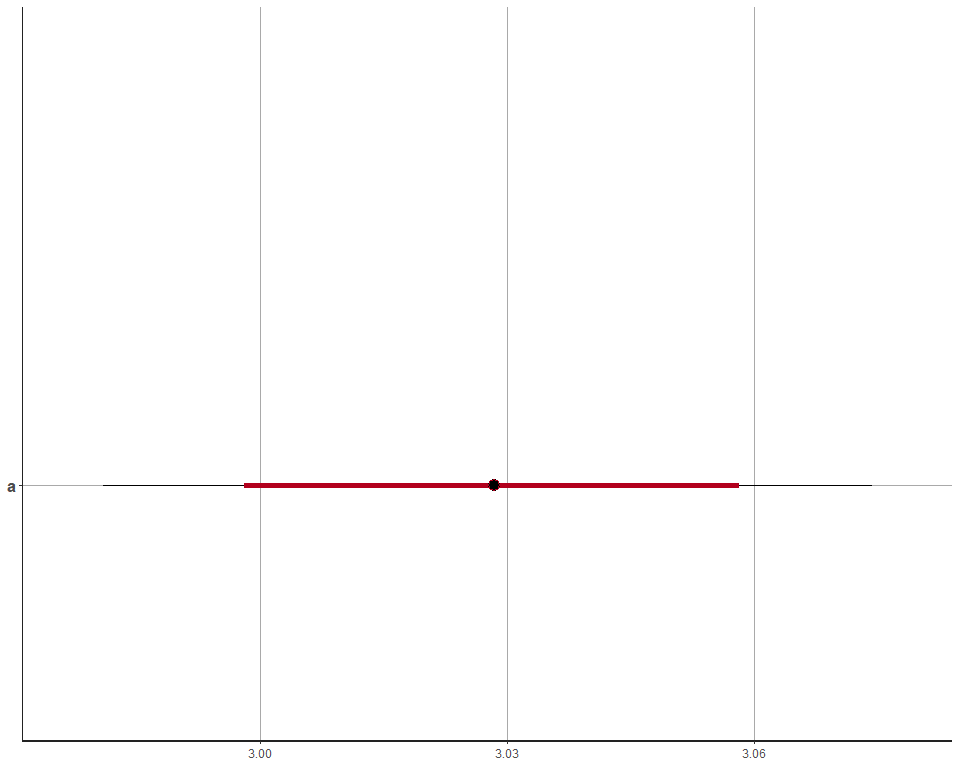
# Model code  
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
}  
parameters{  
 real a;  
}  
model{  
 vector[N] lamda;  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a;  
 }  
 D ~ poisson\_log( lamda );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a;  
 log\_lik[i] = poisson\_log\_lpmf( D[i] | lamda[i] );  
 }  
}  
"  
  
# Fit stan model with model code and model data  
fit.stan.0 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.0 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



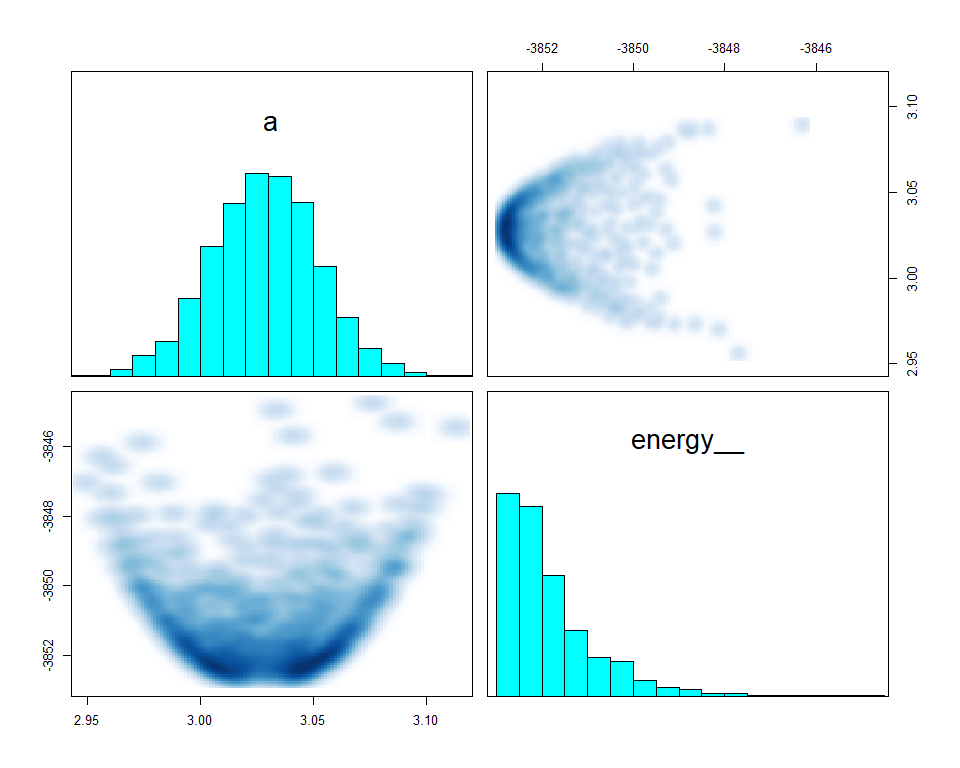
# Show results  
print( fit.stan.0 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 9ebda01fac82b936d9bfeca36a19d78f.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## a 3.03 0 0.02 2.99 3.07 1656 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:21:41 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# Plot coeftab  
plot( fit.stan.0 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



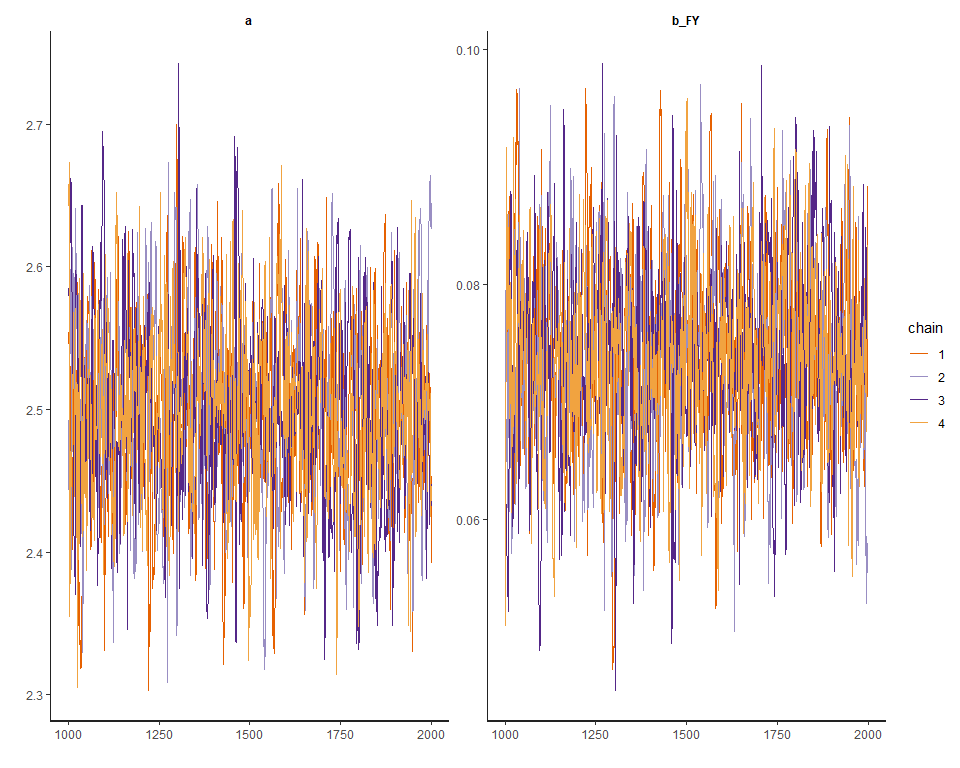
# Pairs plot  
pairs( fit.stan.0 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Draw posterior  
post.0 <- as.data.frame( fit.stan.0 )  
  
# Recreate lamda and simulate it with new data  
f\_lamda <- function(x) exp(post.0$a)  
femininity\_new <- seq( from=min(data$femininity) , to=max(data$femininity) , length.out=30 )  
  
lamda <-   
 sapply(femininity\_new, f\_lamda) %>%  
 as\_tibble() %>%  
 rename\_all(function(x) femininity\_new) %>%  
 mutate(Iter = row\_number()) %>%  
 gather(femininity, deaths, -Iter) %>%  
 group\_by(femininity) %>%  
 mutate(hpdi\_l = quantile(deaths, probs = 0.055),  
 hpdi\_h = quantile(deaths, probs = 0.945)) %>%  
 mutate(mu = mean(deaths)) %>%  
 ungroup() %>%  
 mutate(femininity = as.numeric(femininity))

## Warning: `as\_tibble.matrix()` requires a matrix with column names or a `.name\_repair` argument. Using compatibility `.name\_repair`.  
## This warning is displayed once per session.

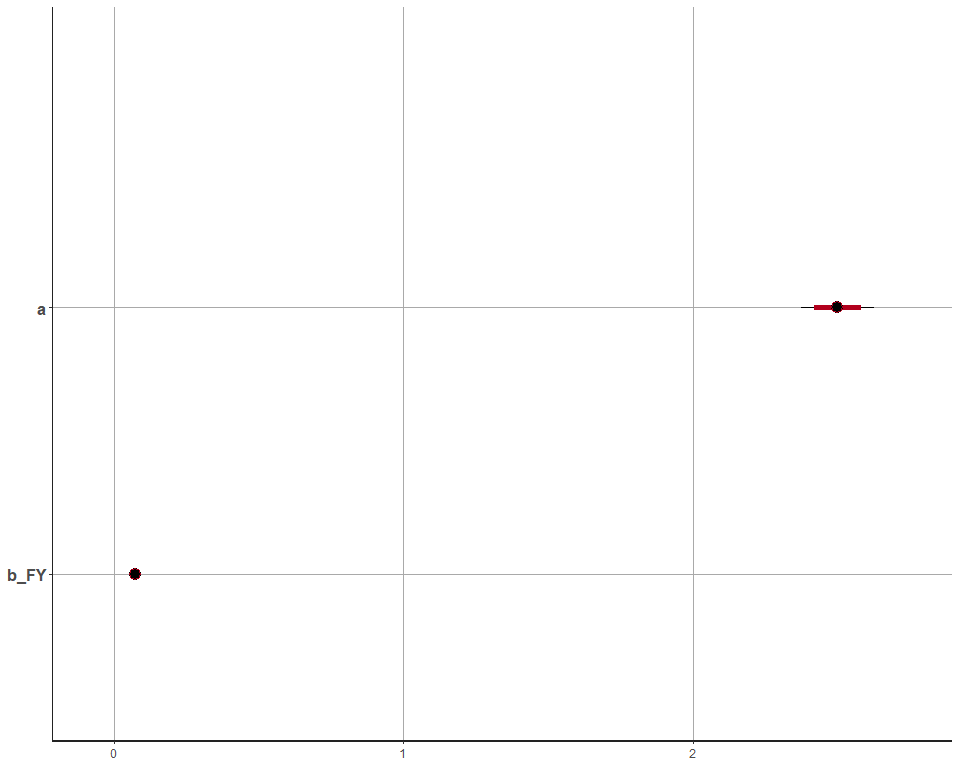
# Plot raw data and model estimate of lamda  
p <- ggplot()   
p0 <- p +   
 geom\_point(data = data,  
 aes(femininity, deaths),   
 shape = 1, color = 'dodgerblue') +  
 geom\_ribbon(data = lamda,  
 aes(x = femininity, ymin = hpdi\_l, ymax = hpdi\_h), alpha = .5) +  
 geom\_line(data = lamda,  
 aes(x = femininity, y = mu)) +  
 ggtitle("Intercept only")  
  
# Model code  
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
}  
parameters{  
 real a;  
 real b\_FY;  
}  
model{  
 vector[N] lamda;  
 b\_FY ~ uniform( -15 , 15 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY \* FY[i];  
 }  
 D ~ poisson\_log( lamda );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY \* FY[i];  
 log\_lik[i] = poisson\_log\_lpmf( D[i] | lamda[i] );  
 }  
}  
"  
  
# Fit stan model with model code and model data  
fit.stan.1 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.1 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



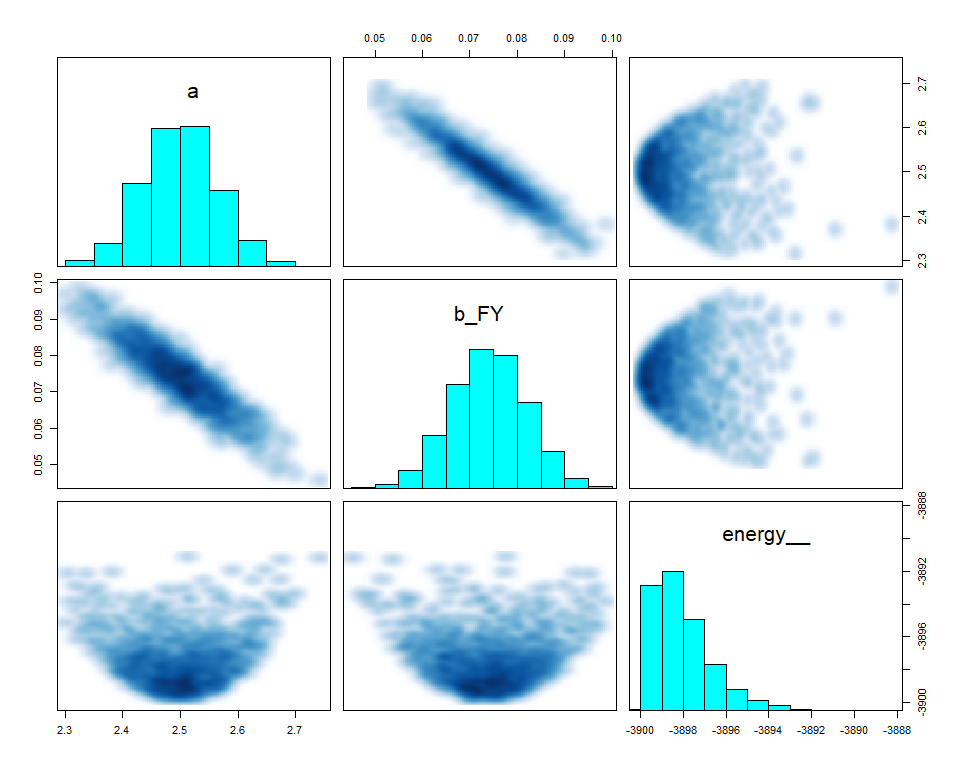
# Show results  
print( fit.stan.1 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: ce66ef3ed6babaf5df357e697b495a94.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## a 2.50 0 0.06 2.40 2.61 970 1  
## b\_FY 0.07 0 0.01 0.06 0.09 983 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:23:29 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

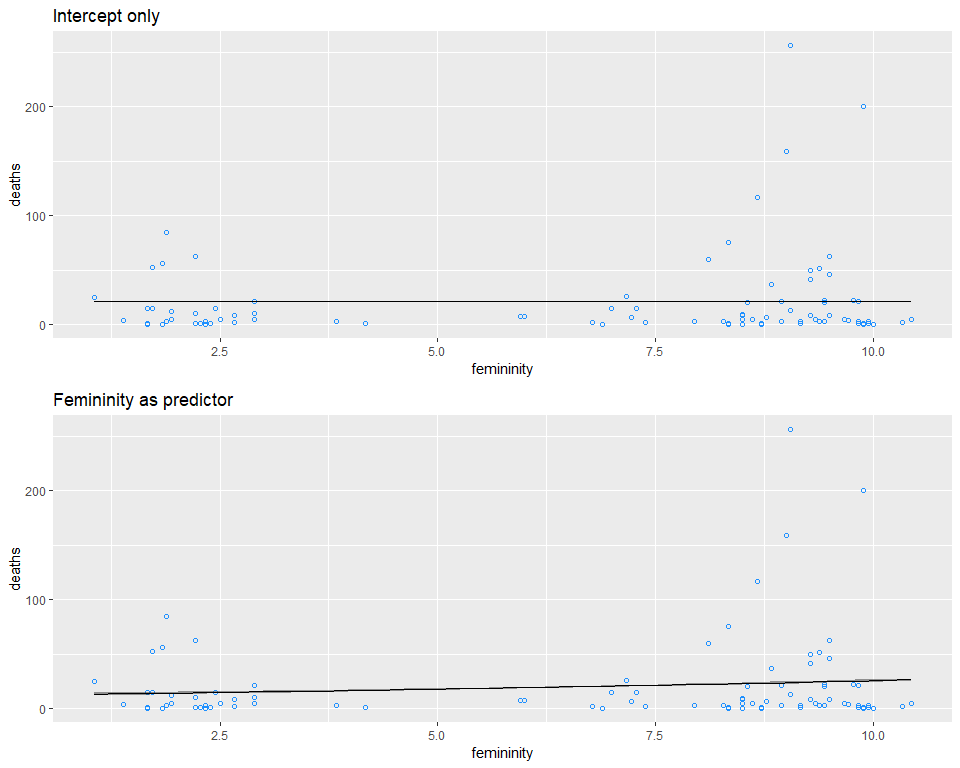
# Plot coeftab  
plot( fit.stan.1 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.1 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Draw posterior  
post.1 <- as.data.frame( fit.stan.1 )  
  
# Recreate lamda and simulate it with new data  
f\_lamda <- function(x) exp(post.1$a + post.1$b\_FY \* x)  
femininity\_new <- seq( from=min(data$femininity) , to=max(data$femininity) , length.out=30 )  
  
lamda <-   
 sapply(femininity\_new, f\_lamda) %>%  
 as\_tibble() %>%  
 rename\_all(function(x) femininity\_new) %>%  
 mutate(Iter = row\_number()) %>%  
 gather(femininity, deaths, -Iter) %>%  
 group\_by(femininity) %>%  
 mutate(hpdi\_l = quantile(deaths, probs = 0.055),  
 hpdi\_h = quantile(deaths, probs = 0.945)) %>%  
 mutate(mu = mean(deaths)) %>%  
 ungroup() %>%  
 mutate(femininity = as.numeric(femininity))  
  
# Plot raw data and model estimate of lamda  
p <- ggplot()   
p1 <- p +   
 geom\_point(data = data,  
 aes(femininity, deaths),   
 shape = 1, color = 'dodgerblue') +  
 geom\_ribbon(data = lamda,  
 aes(x = femininity, ymin = hpdi\_l, ymax = hpdi\_h), alpha = .5) +  
 geom\_line(data = lamda,  
 aes(x = femininity, y = mu)) +  
 ggtitle("Femininity as predictor")  
  
grid.arrange(p0, p1, nrow = 2)

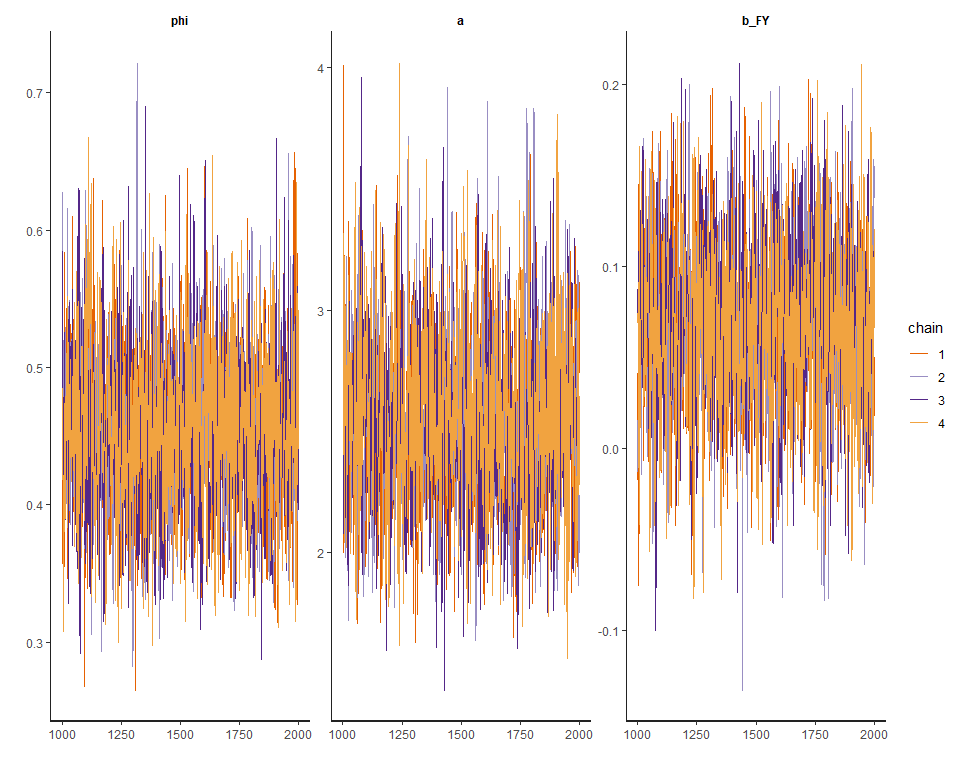


## Conclusion

The association between femininity of names and deaths is quite obvious that if we add femininity as a predictor variable, it helps to predict deaths more properly. However, despite the model fits well on most of the cases, it fit poorly on the cases which have high variation.

# Question 2

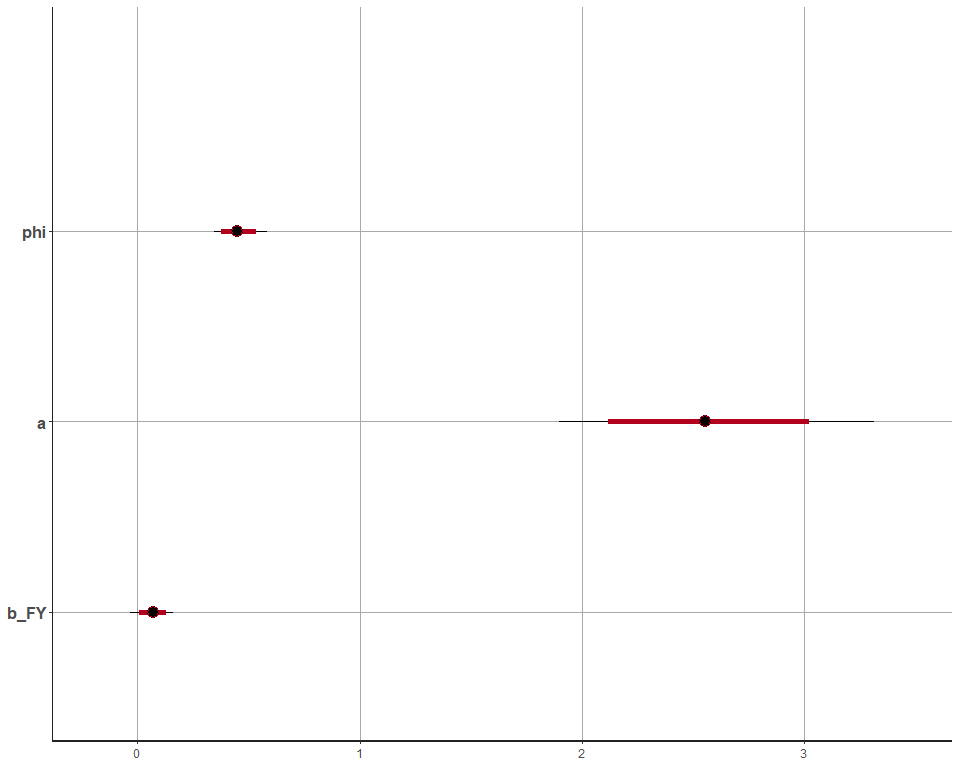
# Model code  
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
}  
parameters{  
 real<lower=0> phi;  
 real a;  
 real b\_FY;  
}  
model{  
 vector[N] lamda;  
 phi ~ cauchy( 0 , 2 );  
 b\_FY ~ uniform( -30 , 30 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY \* FY[i];  
 }  
 D ~ neg\_binomial\_2\_log( lamda , phi );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY \* FY[i];  
 log\_lik[i] = neg\_binomial\_2\_log\_lpmf( D[i] | lamda[i] , phi );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.2 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.2 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



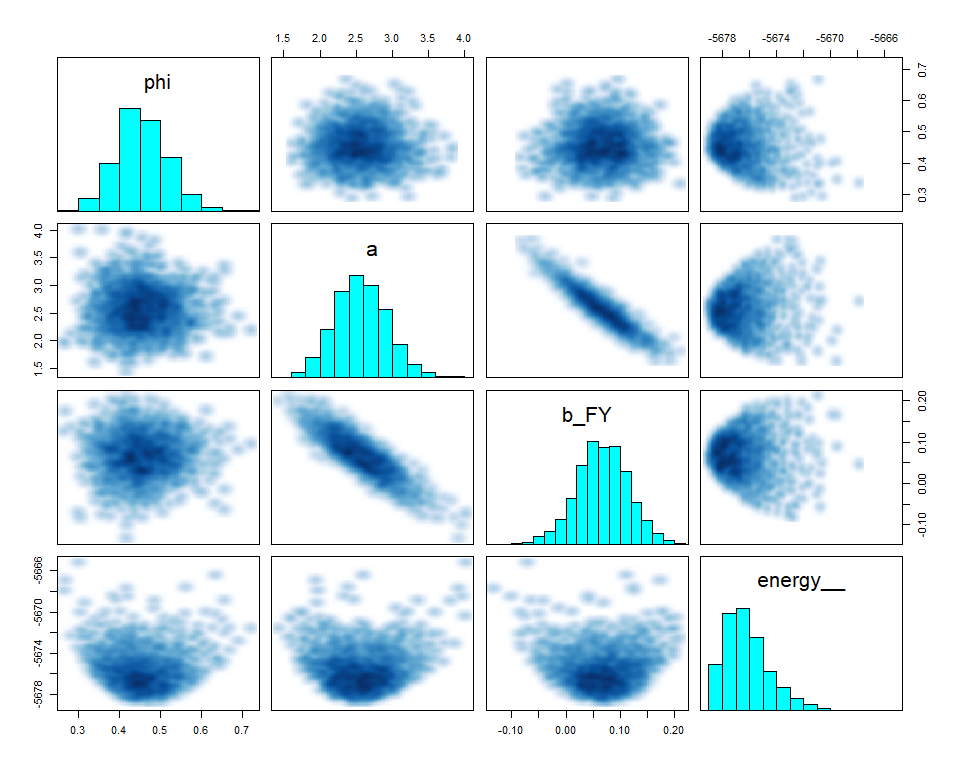
# Show results  
print( fit.stan.2 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 31935db4aab5bf20580fba3f06e6e13d.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## phi 0.45 0.00 0.06 0.36 0.55 2211 1  
## a 2.56 0.01 0.36 2.00 3.14 1801 1  
## b\_FY 0.07 0.00 0.05 -0.01 0.14 1799 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:25:28 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

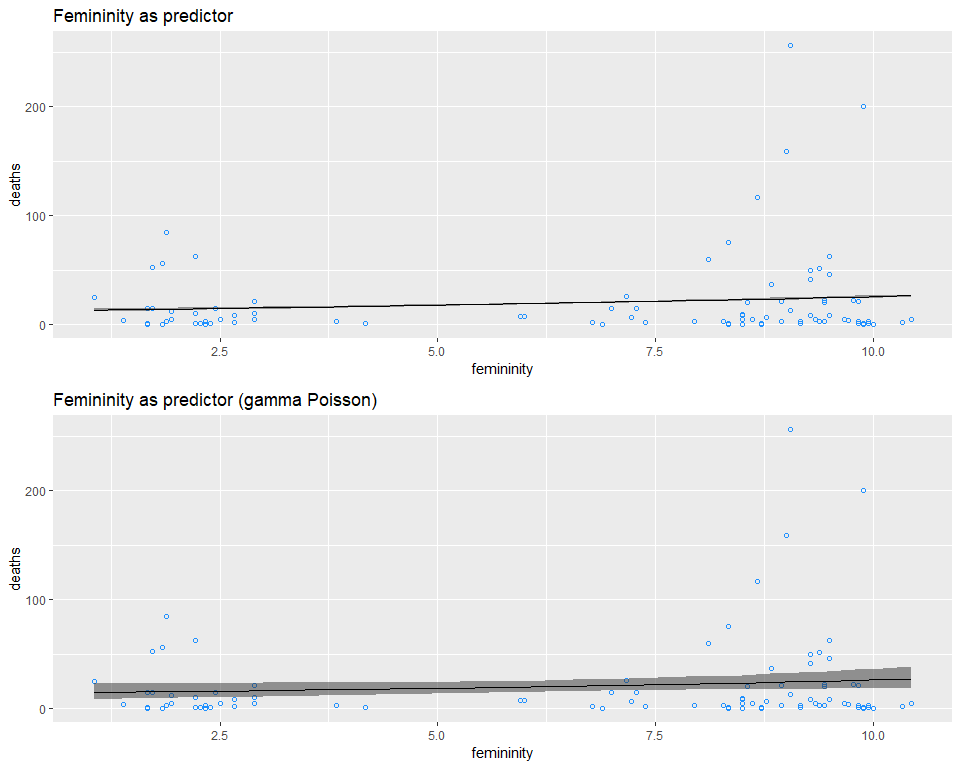
# Plot coeftab  
plot( fit.stan.2 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.2 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Draw posterior  
post.2 <- as.data.frame( fit.stan.2 )  
  
# Recreate lamda and simulate it with new data  
f\_lamda <- function(x) exp(post.2$a + post.2$b\_FY \* x)  
femininity\_new <- seq( from=min(data$femininity) , to=max(data$femininity) , length.out=30 )  
  
lamda <-   
 sapply(femininity\_new, f\_lamda) %>%  
 as\_tibble() %>%  
 rename\_all(function(x) femininity\_new) %>%  
 mutate(Iter = row\_number()) %>%  
 gather(femininity, deaths, -Iter) %>%  
 group\_by(femininity) %>%  
 mutate(hpdi\_l = quantile(deaths, probs = 0.055),  
 hpdi\_h = quantile(deaths, probs = 0.945)) %>%  
 mutate(mu = mean(deaths)) %>%  
 ungroup() %>%  
 mutate(femininity = as.numeric(femininity))  
  
# Plot raw data and model estimate of lamda  
p <- ggplot()   
p2 <- p +   
 geom\_point(data = data,  
 aes(femininity, deaths),   
 shape = 1, color = 'dodgerblue') +  
 geom\_ribbon(data = lamda,  
 aes(x = femininity, ymin = hpdi\_l, ymax = hpdi\_h), alpha = .5) +  
 geom\_line(data = lamda,  
 aes(x = femininity, y = mu)) +  
 ggtitle("Femininity as predictor (gamma Poisson)")  
  
grid.arrange(p1, p2, nrow = 2)



## Conclusion

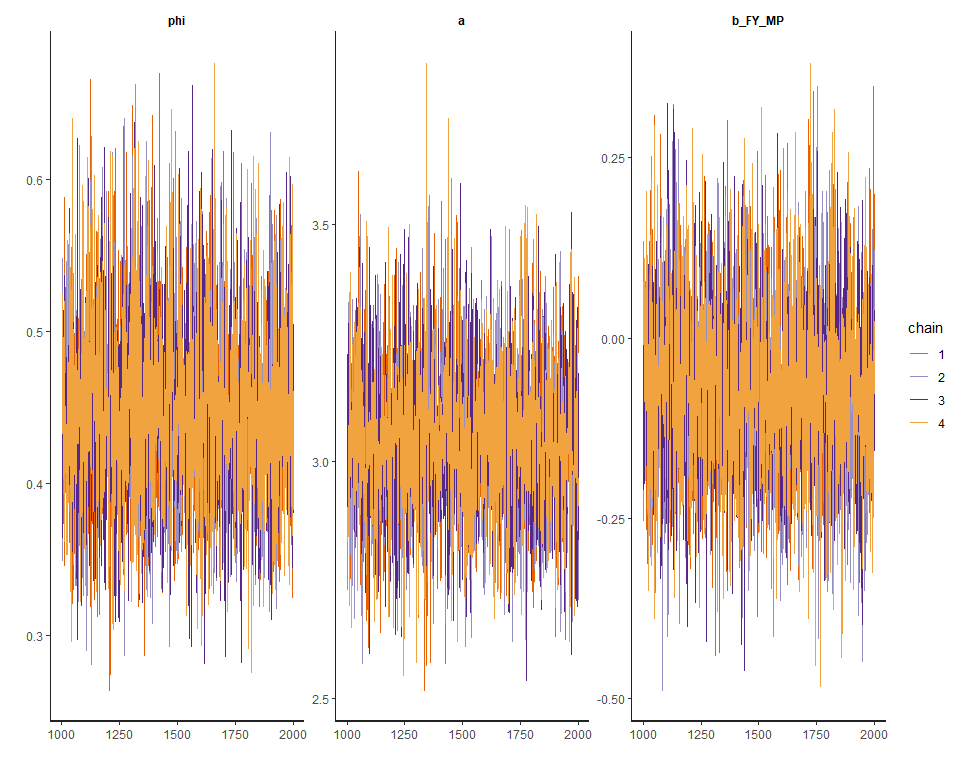
Since the gamma-Poisson distribution assumes that the data has more variation, the coefficient of femininity would have wider range, and thus it overlaps zero.

# Question 3

# Create data list for the stan model  
model.data <- list(  
 N = NROW( data ),  
 D = data$deaths,  
 C = data$category,  
 MP = scale( data$min\_pressure )[ ,1],  
 DN = scale( data$damage\_norm )[ ,1],  
 FE = data$female,  
 FY = scale( data$femininity )[ ,1]  
)  
str( model.data )

## List of 7  
## $ N : int 92  
## $ D : int [1:92] 2 4 3 1 0 60 20 20 0 200 ...  
## $ C : int [1:92] 3 3 1 1 1 3 3 4 3 1 ...  
## $ MP: num [1:92] -0.258 -0.52 1.054 1.158 1.054 ...  
## $ DN: num [1:92] -0.439 -0.148 -0.55 -0.558 -0.561 ...  
## $ FE: int [1:92] 1 0 0 1 1 1 1 1 1 1 ...  
## $ FY: num [1:92] -0.000935 -1.670758 -0.913314 0.94587 0.481074 ...

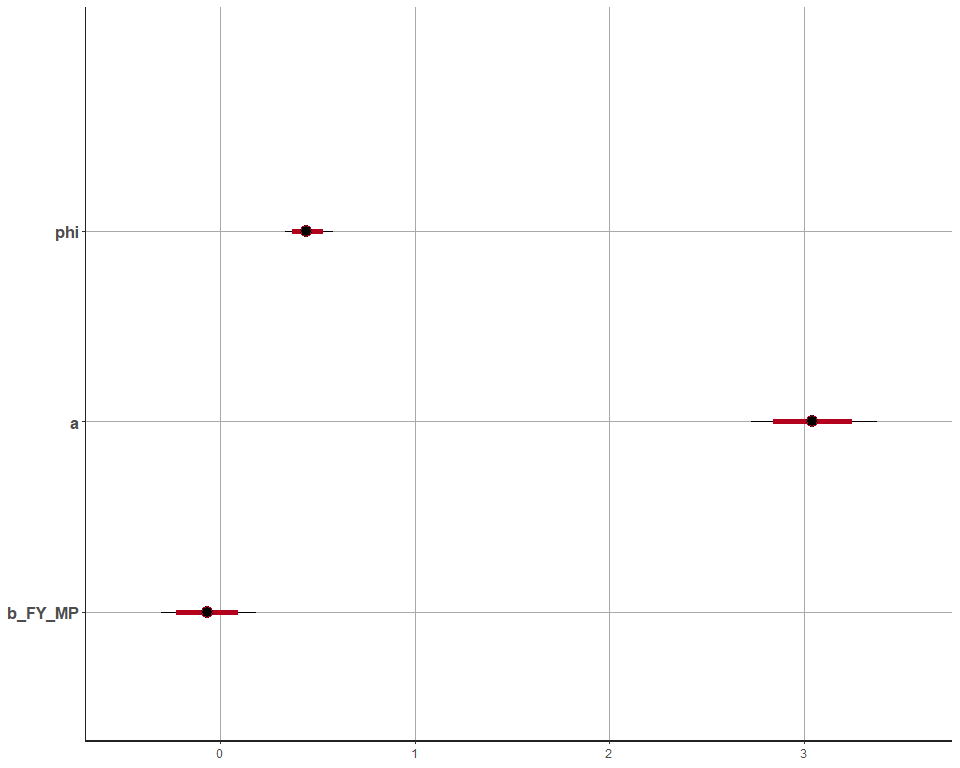
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
 real MP[N];  
}  
parameters{  
 real<lower=0> phi;  
 real a;  
 real b\_FY\_MP;  
}  
model{  
 vector[N] lamda;  
 phi ~ cauchy( 0 , 2 );  
 b\_FY\_MP ~ uniform( -30 , 30 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_MP \* FY[i] \* MP[i];  
 }  
 D ~ neg\_binomial\_2\_log( lamda , phi );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_MP \* FY[i] \* MP[i];  
 log\_lik[i] = neg\_binomial\_2\_log\_lpmf( D[i] | lamda[i] , phi );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.3 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.3 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



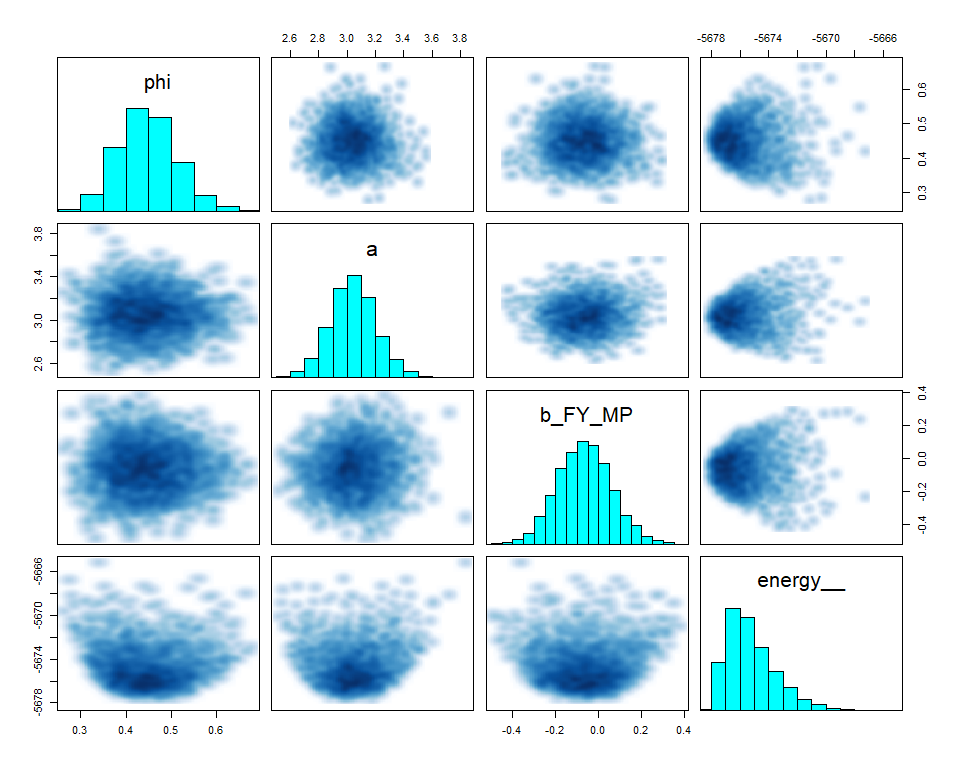
# Show results  
print( fit.stan.3 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: fae55111693c447cac71f7c52c226c54.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## phi 0.45 0 0.06 0.35 0.55 3582 1  
## a 3.04 0 0.16 2.79 3.32 3663 1  
## b\_FY\_MP -0.07 0 0.13 -0.27 0.14 3691 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:27:25 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

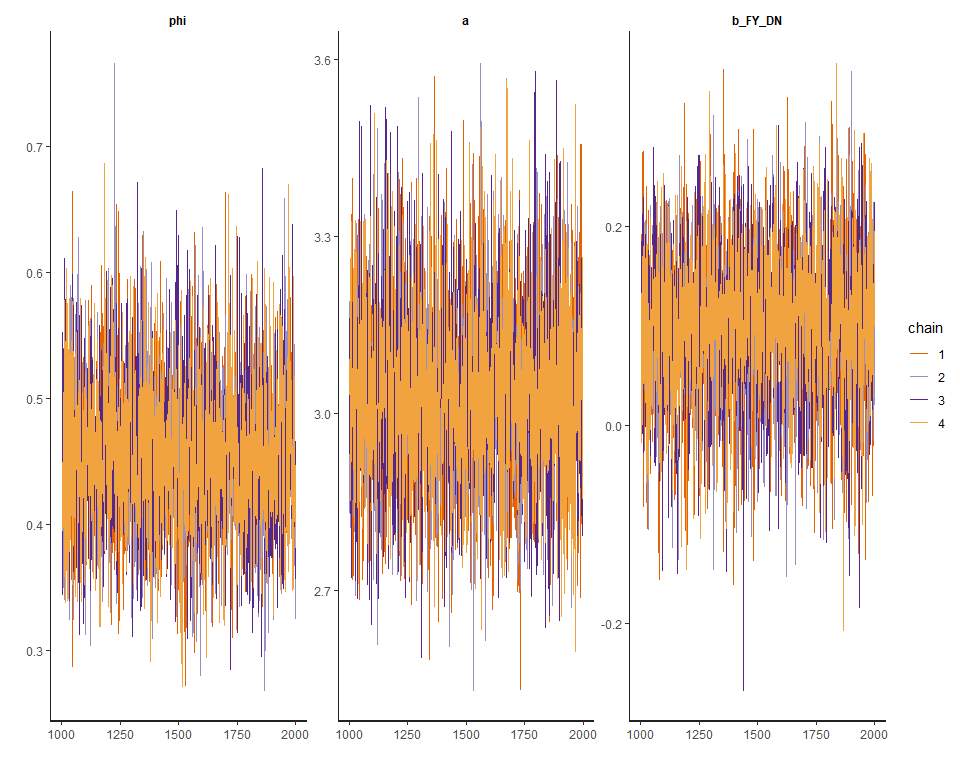
# Plot coeftab  
plot( fit.stan.3 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.3 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



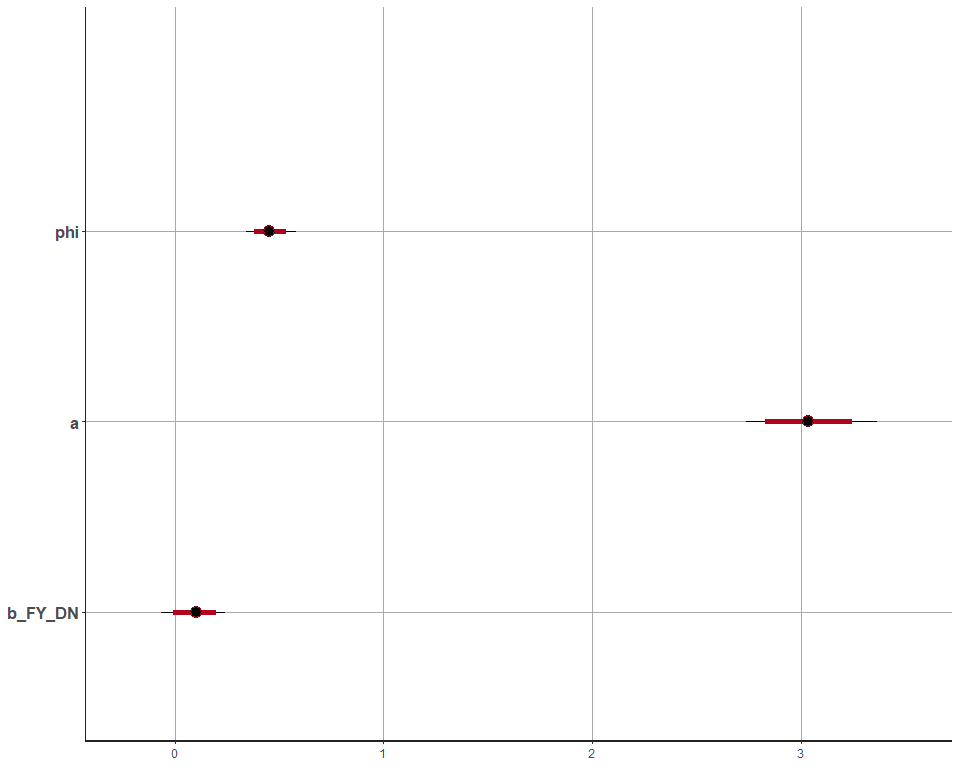
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
 real DN[N];  
}  
parameters{  
 real<lower=0> phi;  
 real a;  
 real b\_FY\_DN;  
}  
model{  
 vector[N] lamda;  
 phi ~ cauchy( 0 , 2 );  
 b\_FY\_DN ~ uniform( -30 , 30 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_DN \* FY[i] \* DN[i];  
 }  
 D ~ neg\_binomial\_2\_log( lamda , phi );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_DN \* FY[i] \* DN[i];  
 log\_lik[i] = neg\_binomial\_2\_log\_lpmf( D[i] | lamda[i] , phi );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.4 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.4 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



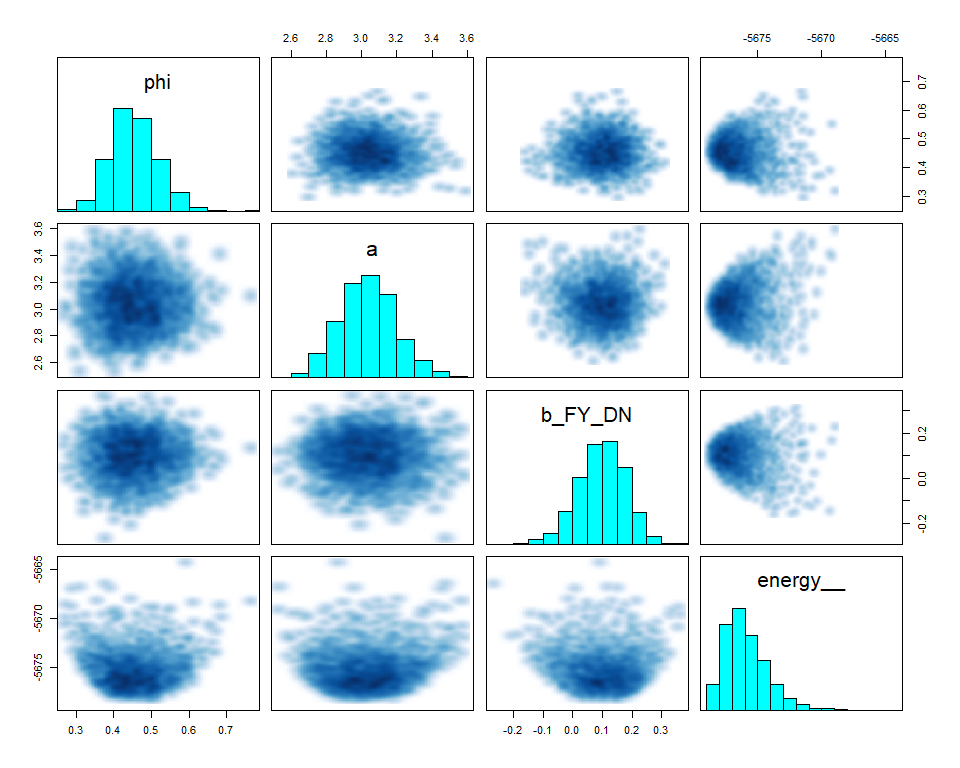
# Show results  
print( fit.stan.4 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 62b873885424d4633e887f3ca7e10c3a.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## phi 0.45 0 0.06 0.36 0.56 3238 1  
## a 3.04 0 0.16 2.79 3.30 3569 1  
## b\_FY\_DN 0.10 0 0.08 -0.04 0.22 3702 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:29:06 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

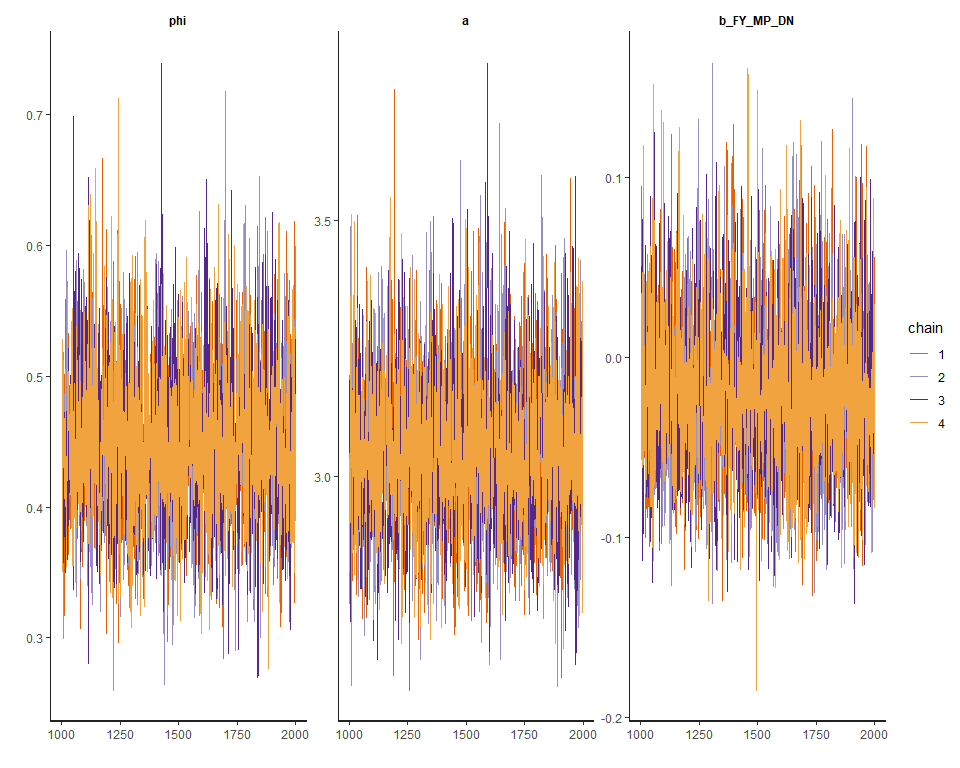
# Plot coeftab  
plot( fit.stan.4 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.4 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



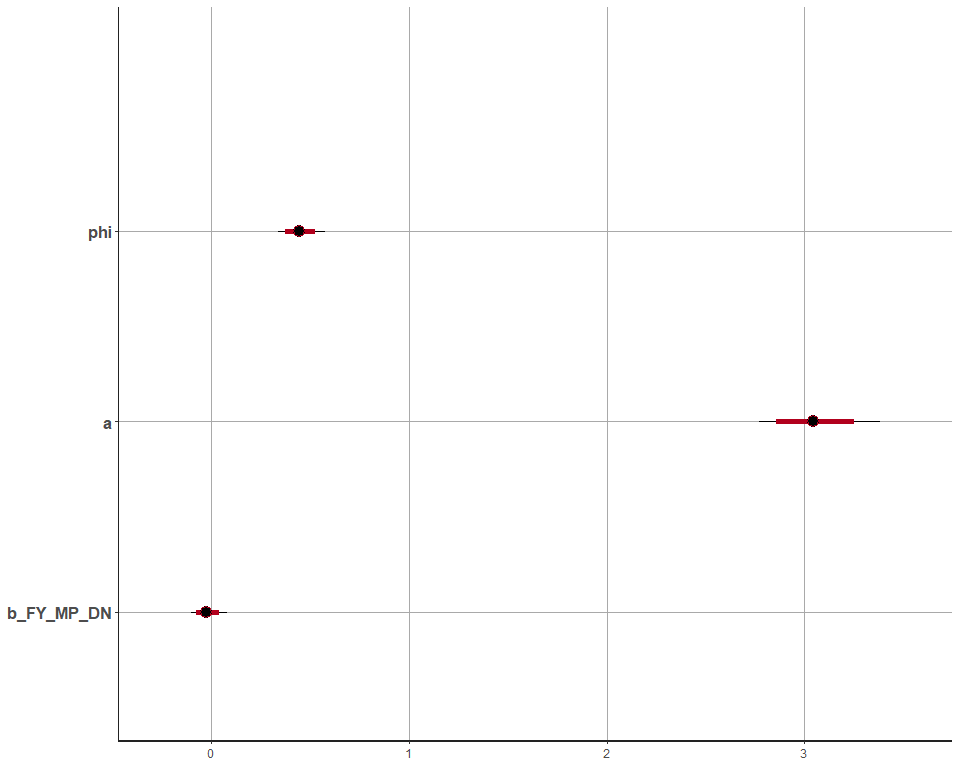
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
 real MP[N];  
 real DN[N];  
}  
parameters{  
 real<lower=0> phi;  
 real a;  
 real b\_FY\_MP\_DN;  
}  
model{  
 vector[N] lamda;  
 phi ~ cauchy( 0 , 2 );  
 b\_FY\_MP\_DN ~ uniform( -30 , 30 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_MP\_DN \* FY[i] \* MP[i] \* DN[i];  
 }  
 D ~ neg\_binomial\_2\_log( lamda , phi );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_MP\_DN \* FY[i] \* MP[i] \* DN[i];  
 log\_lik[i] = neg\_binomial\_2\_log\_lpmf( D[i] | lamda[i] , phi );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.5 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.5 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



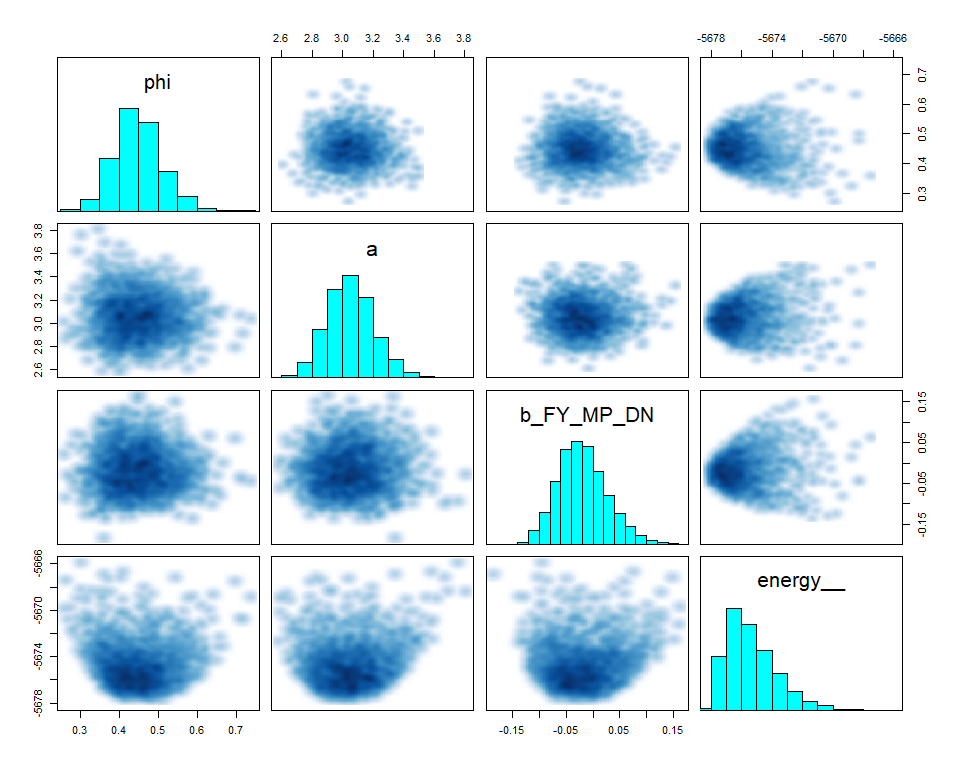
# Show results  
print( fit.stan.5 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 0f15ef1a30274481a62569cd36897069.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## phi 0.45 0 0.06 0.36 0.55 3442 1  
## a 3.05 0 0.16 2.81 3.31 3643 1  
## b\_FY\_MP\_DN -0.02 0 0.05 -0.09 0.06 3206 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:30:51 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# Plot coeftab  
plot( fit.stan.5 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.5 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Calculate WAIC and compare models  
log\_lik\_3 <- extract\_log\_lik(fit.stan.3, merge\_chains = FALSE)  
waic\_3 <- waic(log\_lik\_3)

## Warning: 2 (2.2%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

log\_lik\_4 <- extract\_log\_lik(fit.stan.4, merge\_chains = FALSE)  
waic\_4 <- waic(log\_lik\_4)

## Warning: 4 (4.3%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

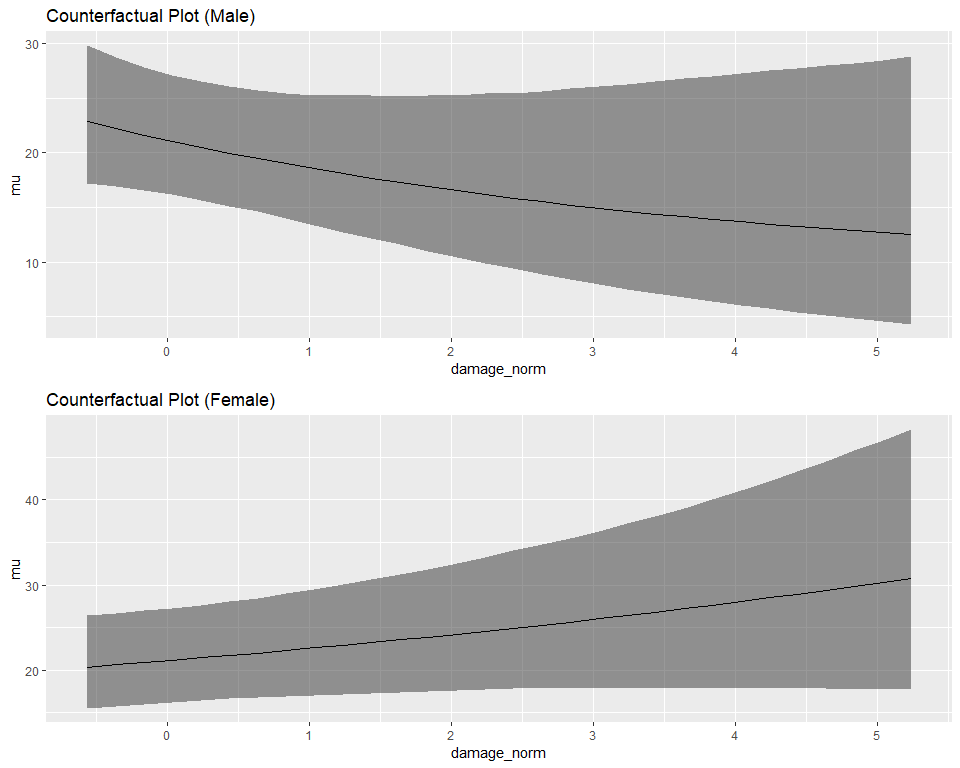
log\_lik\_5 <- extract\_log\_lik(fit.stan.5, merge\_chains = FALSE)  
waic\_5 <- waic(log\_lik\_5)

## Warning: 4 (4.3%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

compare\_model <- loo::compare(waic\_3, waic\_4, waic\_5)  
compare\_model

## elpd\_diff se\_diff elpd\_waic p\_waic waic   
## waic\_4 0.0 0.0 -355.7 4.5 711.4  
## waic\_5 -0.4 1.1 -356.1 4.1 712.1  
## waic\_3 -0.8 2.1 -356.5 4.8 713.0

# Draw posterior  
post.4 <- as.data.frame( fit.stan.4 )  
data$femininity\_scale <- scale( data$femininity )[ , 1]  
  
# Recreate lamda and simulate it with new data (Male)  
f\_lamda <- function(x) exp(post.4$a + post.4$b\_FY\_DN \* mean(data[ data$female==0 , ]$femininity\_scale) \* x)  
damage\_norm\_scale <- scale(data$damage\_norm)[,1]  
damage\_norm\_new <- seq( from=min(damage\_norm\_scale) , to=max(damage\_norm\_scale) , length.out=30 )  
  
lamda <-   
 sapply(damage\_norm\_new, f\_lamda) %>%  
 as\_tibble() %>%  
 rename\_all(function(x) damage\_norm\_new) %>%  
 mutate(Iter = row\_number()) %>%  
 gather(damage\_norm, deaths, -Iter) %>%  
 group\_by(damage\_norm) %>%  
 mutate(hpdi\_l = quantile(deaths, probs = 0.055),  
 hpdi\_h = quantile(deaths, probs = 0.945)) %>%  
 mutate(mu = mean(deaths)) %>%  
 ungroup() %>%  
 mutate(damage\_norm = as.numeric(damage\_norm))  
  
# Plot raw data and model estimate of lamda  
p <- ggplot()   
p4\_m <- p +   
 geom\_ribbon(data = lamda,  
 aes(x = damage\_norm, ymin = hpdi\_l, ymax = hpdi\_h), alpha = .5) +  
 geom\_line(data = lamda,  
 aes(x = damage\_norm, y = mu)) +  
 ggtitle("Counterfactual Plot (Male)")  
  
# Recreate lamda and simulate it with new data (Female)  
f\_lamda <- function(x) exp(post.4$a + post.4$b\_FY\_DN \* mean(data[ data$female==1 , ]$femininity\_scale) \* x)  
damage\_norm\_scale <- scale(data$damage\_norm)[,1]  
damage\_norm\_new <- seq( from=min(damage\_norm\_scale) , to=max(damage\_norm\_scale) , length.out=30 )  
  
lamda <-   
 sapply(damage\_norm\_new, f\_lamda) %>%  
 as\_tibble() %>%  
 rename\_all(function(x) damage\_norm\_new) %>%  
 mutate(Iter = row\_number()) %>%  
 gather(damage\_norm, deaths, -Iter) %>%  
 group\_by(damage\_norm) %>%  
 mutate(hpdi\_l = quantile(deaths, probs = 0.055),  
 hpdi\_h = quantile(deaths, probs = 0.945)) %>%  
 mutate(mu = mean(deaths)) %>%  
 ungroup() %>%  
 mutate(damage\_norm = as.numeric(damage\_norm))  
  
# Plot raw data and model estimate of lamda  
p <- ggplot()   
p4\_f <- p +   
 geom\_ribbon(data = lamda,  
 aes(x = damage\_norm, ymin = hpdi\_l, ymax = hpdi\_h), alpha = .5) +  
 geom\_line(data = lamda,  
 aes(x = damage\_norm, y = mu)) +  
 ggtitle("Counterfactual Plot (Female)")  
  
grid.arrange(p4\_m, p4\_f, nrow = 2)



## Conclusion

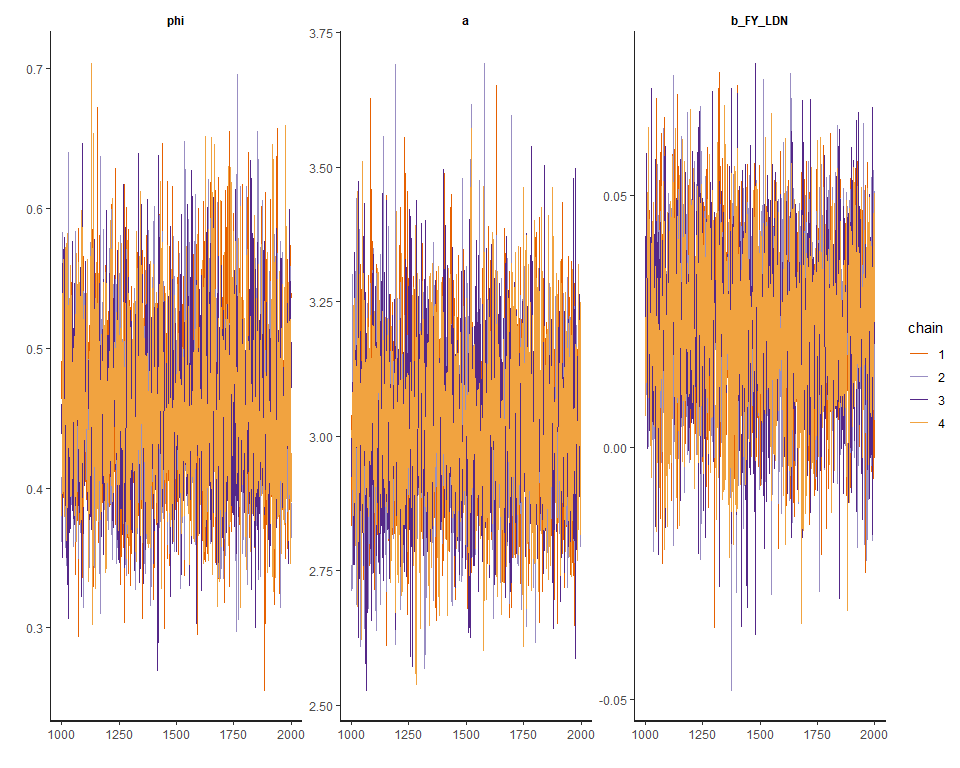
By comparing WAIC, we could know that the model which considers the interaction between femininity and damage\_norm is the mmost plausible. Besides, from the counterfactual prediction plots, we also found that the effect sizes are quite obvious when contrasting the predictions of hurricanes with masculine and feminine names.

# Question 4

# Calculate logarithm of damage\_norm  
data$log\_damage\_norm <- log( data$damage\_norm )  
  
# Create data list for the stan model  
model.data <- list(  
 N = NROW( data ),  
 D = data$deaths,  
 C = data$category,  
 MP = scale( data$min\_pressure )[,1],  
 DN = scale( data$damage\_norm )[,1],  
 FE = data$female,  
 FY = scale( data$femininity )[ ,1],  
 LDN = data$log\_damage\_norm  
)  
str( model.data )

## List of 8  
## $ N : int 92  
## $ D : int [1:92] 2 4 3 1 0 60 20 20 0 200 ...  
## $ C : int [1:92] 3 3 1 1 1 3 3 4 3 1 ...  
## $ MP : num [1:92] -0.258 -0.52 1.054 1.158 1.054 ...  
## $ DN : num [1:92] -0.439 -0.148 -0.55 -0.558 -0.561 ...  
## $ FE : int [1:92] 1 0 0 1 1 1 1 1 1 1 ...  
## $ FY : num [1:92] -0.000935 -1.670758 -0.913314 0.94587 0.481074 ...  
## $ LDN: num [1:92] 7.37 8.58 5.01 4.06 2.71 ...

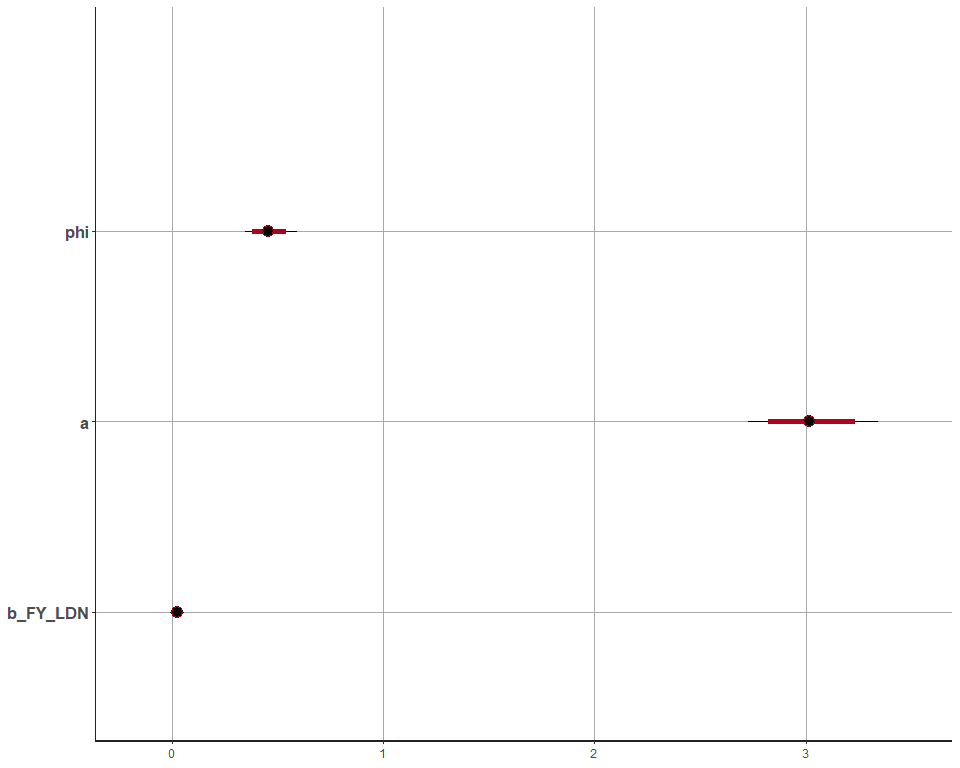
stan.model <- "  
data{  
 int<lower=1> N;  
 int D[N];  
 real FY[N];  
 real LDN[N];  
}  
parameters{  
 real<lower=0> phi;  
 real a;  
 real b\_FY\_LDN;  
}  
model{  
 vector[N] lamda;  
 phi ~ cauchy( 0 , 2 );  
 b\_FY\_LDN ~ uniform( -30 , 30 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_LDN \* FY[i] \* LDN[i];  
 }  
 D ~ neg\_binomial\_2\_log( lamda , phi );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_FY\_LDN \* FY[i] \* LDN[i];  
 log\_lik[i] = neg\_binomial\_2\_log\_lpmf( D[i] | lamda[i] , phi );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.6 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Traceplot  
traceplot( fit.stan.6 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



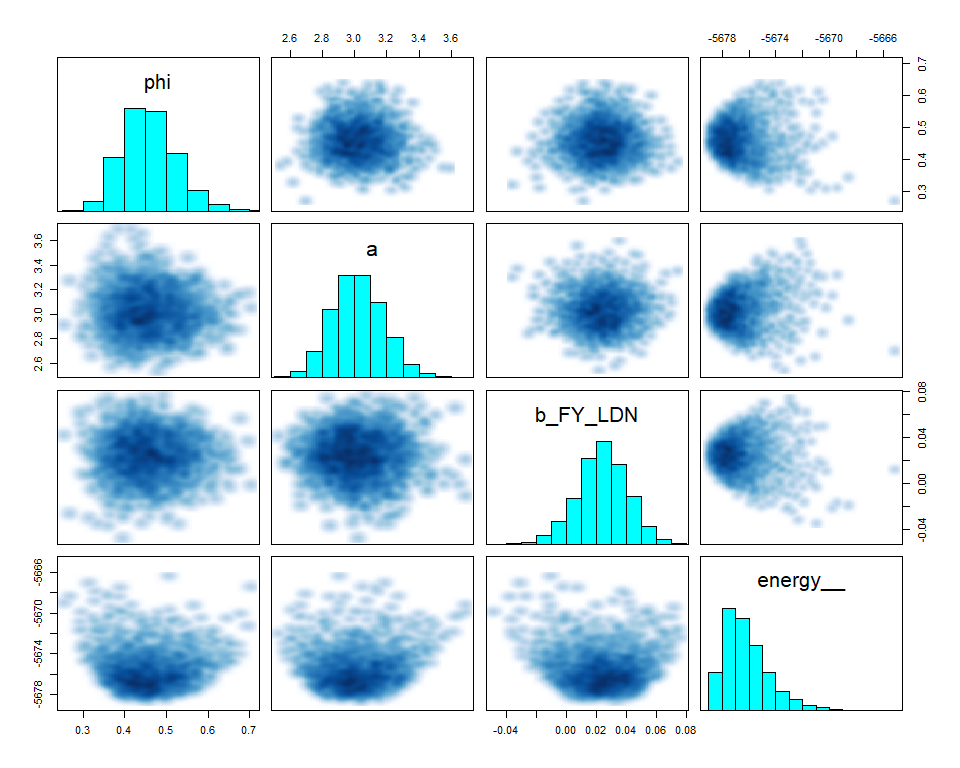
# Show results  
print( fit.stan.6 , probs=c(0.055,0.945) , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 085427fa400cff4872d359909a1334c4.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## phi 0.46 0 0.06 0.36 0.57 3558 1  
## a 3.02 0 0.16 2.78 3.28 3741 1  
## b\_FY\_LDN 0.02 0 0.02 0.00 0.05 3635 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:33:05 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# Plot coeftab  
plot( fit.stan.6 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Pairs plot  
pairs( fit.stan.6 , pars=c("log\_lik","lamda","lp\_\_"), include=FALSE )



# Calculate WAIC and compare models  
log\_lik\_6 <- extract\_log\_lik(fit.stan.6, merge\_chains = FALSE)  
waic\_6 <- waic(log\_lik\_6)

## Warning: 3 (3.3%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

compare\_model <- loo::compare(waic\_4, waic\_6)  
compare\_model

## elpd\_diff se   
## 0.9 2.0

## Conclusion

By comparing WAIC values, since the elpd\_diff value is negative, we should consider that the model fit.stan.4 is better. It implies that mortality increases linearly with a linear increase in storm strength. Thus, when predicting deaths, we should use damage\_norm directly instead of logarithm.

# Question 5

# Load data  
data( "bangladesh" )  
data <- bangladesh; rm( bangladesh )  
str( data )

## 'data.frame': 1934 obs. of 6 variables:  
## $ woman : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ district : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ use.contraception: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ living.children : int 4 1 3 4 1 1 4 4 2 4 ...  
## $ age.centered : num 18.44 -5.56 1.44 8.44 -13.56 ...  
## $ urban : int 1 1 1 1 1 1 1 1 1 1 ...

sort(unique(data$district))

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
## [47] 47 48 49 50 51 52 53 55 56 57 58 59 60 61

data$district\_id <- as.integer( as.factor( data$district ) )  
sort(unique(data$district\_id))

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
## [47] 47 48 49 50 51 52 53 54 55 56 57 58 59 60

# Create data list for the stan model  
model.data <- list(  
 N = NROW( data ),  
 N\_D = max(data$district\_id),  
 W = data$woman,  
 UC = data$use.contraception,  
 LC = data$living.children,  
 AC = scale( data$age.centered )[ ,1],  
 U = data$urban,  
 D = data$district\_id  
)  
str( model.data )

## List of 8  
## $ N : int 1934  
## $ N\_D: int 60  
## $ W : int [1:1934] 1 2 3 4 5 6 7 8 9 10 ...  
## $ UC : int [1:1934] 0 0 0 0 0 0 0 0 0 0 ...  
## $ LC : int [1:1934] 4 1 3 4 1 1 4 4 2 4 ...  
## $ AC : num [1:1934] 2.046 -0.617 0.16 0.936 -1.505 ...  
## $ U : int [1:1934] 1 1 1 1 1 1 1 1 1 1 ...  
## $ D : int [1:1934] 1 1 1 1 1 1 1 1 1 1 ...

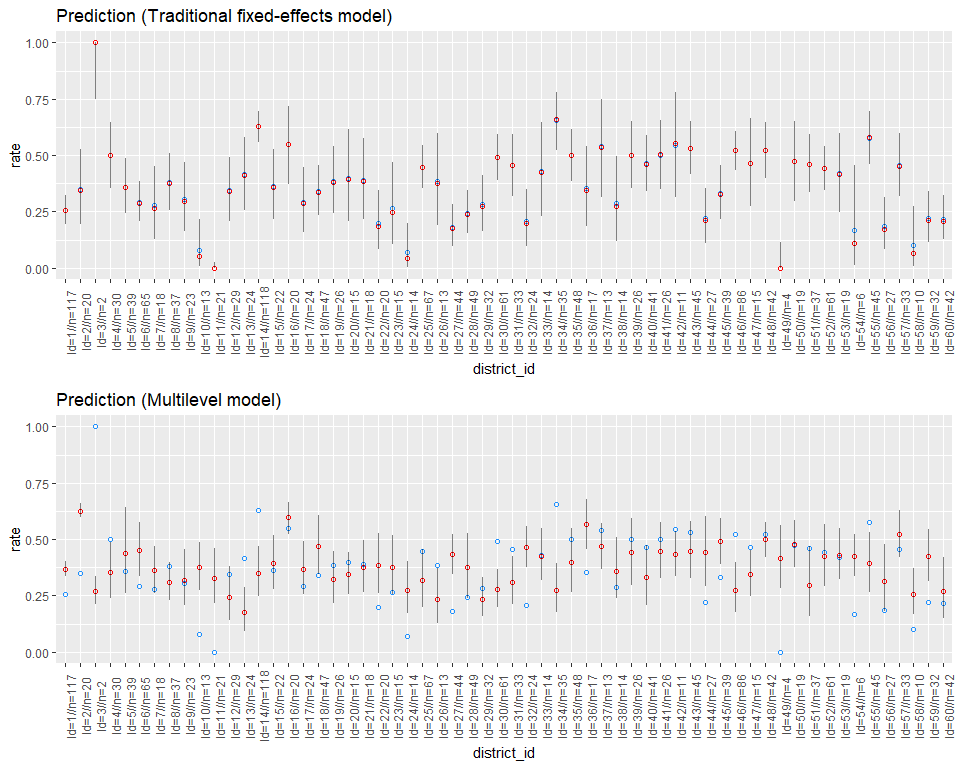
# Traditional fixed-effects model  
stan.model <- "  
data{  
 int<lower=1> N;  
 int<lower=1> N\_D;  
 int D[N];  
 int UC[N];  
}  
parameters{  
 vector[N\_D] a;  
}  
model{  
 vector[N] p;  
 a ~ normal( 0 , 10 );  
 for ( i in 1:N ) {  
 p[i] = a[D[i]];  
 }  
 UC ~ binomial\_logit( 1 , p );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] p;  
 for ( i in 1:N ) {  
 p[i] = a[D[i]];  
 log\_lik[i] = binomial\_logit\_lpmf( UC[i] | 1 , p[i] );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.7 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Show results  
print( fit.stan.7 , probs=c(0.055,0.945) , pars=c("p","log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: a5f9a90cd7725725687bb415826acb71.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## a[1] -1.07 0.00 0.22 -1.42 -0.74 7083 1  
## a[2] -0.65 0.01 0.48 -1.42 0.11 6105 1  
## a[3] 8.56 0.11 5.84 1.08 19.10 3075 1  
## a[4] 0.00 0.00 0.37 -0.60 0.60 7159 1  
## a[5] -0.59 0.00 0.35 -1.15 -0.06 5380 1  
## a[6] -0.90 0.00 0.28 -1.34 -0.47 6990 1  
## a[7] -1.02 0.01 0.55 -1.91 -0.19 7011 1  
## a[8] -0.51 0.00 0.35 -1.06 0.04 6942 1  
## a[9] -0.86 0.01 0.48 -1.64 -0.13 7530 1  
## a[10] -2.93 0.02 1.25 -5.16 -1.28 3482 1  
## a[11] -10.36 0.11 5.53 -20.96 -3.57 2633 1  
## a[12] -0.67 0.00 0.40 -1.34 -0.03 6867 1  
## a[13] -0.35 0.01 0.43 -1.07 0.32 7169 1  
## a[14] 0.52 0.00 0.19 0.22 0.84 5951 1  
## a[15] -0.58 0.01 0.44 -1.29 0.10 6160 1  
## a[16] 0.20 0.01 0.46 -0.52 0.94 6102 1  
## a[17] -0.92 0.01 0.46 -1.66 -0.21 5735 1  
## a[18] -0.68 0.00 0.32 -1.20 -0.18 6967 1  
## a[19] -0.49 0.00 0.41 -1.14 0.16 7389 1  
## a[20] -0.43 0.01 0.56 -1.35 0.48 7131 1  
## a[21] -0.47 0.01 0.50 -1.30 0.30 5645 1  
## a[22] -1.48 0.01 0.57 -2.41 -0.64 5407 1  
## a[23] -1.10 0.01 0.63 -2.16 -0.12 5738 1  
## a[24] -3.07 0.02 1.29 -5.41 -1.41 3605 1  
## a[25] -0.21 0.00 0.25 -0.60 0.17 7619 1  
## a[26] -0.51 0.01 0.59 -1.46 0.40 7227 1  
## a[27] -1.55 0.01 0.41 -2.23 -0.94 5594 1  
## a[28] -1.16 0.00 0.34 -1.70 -0.64 6158 1  
## a[29] -0.98 0.01 0.40 -1.65 -0.36 6117 1  
## a[30] -0.03 0.00 0.26 -0.44 0.38 7345 1  
## a[31] -0.19 0.00 0.36 -0.77 0.37 7299 1  
## a[32] -1.40 0.01 0.51 -2.26 -0.61 6018 1  
## a[33] -0.30 0.01 0.57 -1.21 0.59 6471 1  
## a[34] 0.66 0.00 0.36 0.10 1.25 6304 1  
## a[35] 0.00 0.00 0.29 -0.47 0.46 6727 1  
## a[36] -0.64 0.01 0.52 -1.49 0.17 6647 1  
## a[37] 0.14 0.01 0.59 -0.78 1.09 7391 1  
## a[38] -0.98 0.01 0.63 -2.02 -0.02 6079 1  
## a[39] 0.00 0.00 0.38 -0.61 0.62 6462 1  
## a[40] -0.15 0.00 0.32 -0.67 0.36 7143 1  
## a[41] 0.01 0.00 0.40 -0.63 0.65 6688 1  
## a[42] 0.22 0.01 0.64 -0.79 1.25 5648 1  
## a[43] 0.13 0.00 0.31 -0.35 0.62 6735 1  
## a[44] -1.32 0.01 0.47 -2.11 -0.59 6031 1  
## a[45] -0.72 0.00 0.35 -1.28 -0.18 5837 1  
## a[46] 0.09 0.00 0.22 -0.26 0.43 7510 1  
## a[47] -0.15 0.01 0.51 -0.99 0.67 7573 1  
## a[48] 0.09 0.00 0.31 -0.41 0.60 6510 1  
## a[49] -9.13 0.11 5.83 -19.90 -2.05 2793 1  
## a[50] -0.11 0.01 0.47 -0.86 0.63 7856 1  
## a[51] -0.17 0.00 0.33 -0.69 0.37 6635 1  
## a[52] -0.24 0.00 0.26 -0.64 0.16 7643 1  
## a[53] -0.34 0.01 0.48 -1.12 0.40 6896 1  
## a[54] -2.08 0.03 1.39 -4.47 -0.18 2859 1  
## a[55] 0.32 0.00 0.31 -0.17 0.82 6527 1  
## a[56] -1.56 0.01 0.51 -2.43 -0.79 5828 1  
## a[57] -0.19 0.00 0.36 -0.77 0.39 7429 1  
## a[58] -2.64 0.02 1.27 -4.91 -0.98 3441 1  
## a[59] -1.33 0.01 0.44 -2.05 -0.66 6927 1  
## a[60] -1.33 0.00 0.38 -1.94 -0.74 7315 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:36:18 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# Calculate contraception use rate of original data  
data.con.rate <- data %>%   
 group\_by(district\_id) %>%   
 summarise(  
 count=n(),   
 total\_use\_contraception=sum(use.contraception),  
 rate=total\_use\_contraception/count  
 ) %>%   
 as.data.frame() %>%  
 mutate(  
 x\_label = reorder(as.factor(paste0('Id=',district\_id,'//n=',count)), count)  
 )  
  
# Draw posterior  
post.7 <- as.data.frame( fit.stan.7 )[,1:60]  
  
p\_mean <- as.data.frame( logistic( apply( post.7 , 2 , mean ) ) )  
colnames(p\_mean) <- c("mu")  
p\_mean$district\_id <- seq( from=1 , to=60 )  
p\_mean <- as.data.frame( p\_mean )  
  
p\_PI <- t( as.data.frame( logistic(apply( post.7 , 2, PI , prob=0.89 ) ) ) )  
district\_id <- seq( from=1 , to=60 )  
p\_PI <- cbind( p\_PI , newColumn=district\_id )  
colnames(p\_PI) <- c("hpdi\_l","hpdi\_h","district\_id")  
p\_PI <- as.data.frame( p\_PI )  
  
# Plot raw data and model estimate of each district  
p <- ggplot()   
p7 <- p +   
 geom\_point(data = data.con.rate,  
 aes(district\_id, rate),   
 shape = 1, color = 'dodgerblue') +  
 scale\_x\_discrete(limits = data.con.rate$x\_label) +   
 theme(axis.text.x = element\_text(angle = 90)) +  
 geom\_point(data = p\_mean,  
 aes(district\_id, mu),   
 shape = 1, color = 'red') +  
 geom\_segment(data = p\_PI,  
 aes(x = district\_id, y = hpdi\_l, xend = district\_id, yend = hpdi\_h), alpha = .5) +  
 ggtitle("Prediction (Traditional fixed-effects model)")  
  
# Multilevel model  
stan.model <- "  
data{  
 int<lower=1> N;  
 int<lower=1> N\_D;  
 int D[N];  
 int UC[N];  
}  
parameters{  
 real mu;  
 real<lower=0> sigma;  
 vector[N\_D] a;  
}  
model{  
 vector[N] p;  
 sigma ~ exponential( 1 );  
 mu ~ normal( 0 , 10 );  
 a ~ normal( mu , sigma );  
 for ( i in 1:N ) {  
 p[i] = a[D[i]];  
 }  
 UC ~ binomial\_logit( 1 , p );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] p;  
 for ( i in 1:N ) {  
 p[i] = a[D[i]];  
 log\_lik[i] = binomial\_logit\_lpmf( UC[i] | 1 , p[i] );  
 }  
}  
"  
# Fit stan model with model code and model data  
fit.stan.8 <- stan( model\_code=stan.model , data=model.data ,   
 chains=4 , cores=4 , iter=2000 )  
  
# Show results  
print( fit.stan.8 , probs=c(0.055,0.945) , pars=c("p","log\_lik","lamda","lp\_\_"), include=FALSE )

## Inference for Stan model: 468021ab9d116ed8281cc6ebb7b9dcf0.  
## 4 chains, each with iter=2000; warmup=1000; thin=1;   
## post-warmup draws per chain=1000, total post-warmup draws=4000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## mu -0.54 0.00 0.09 -0.68 -0.40 3695 1  
## sigma 0.52 0.00 0.08 0.39 0.65 1493 1  
## a[1] -0.99 0.00 0.20 -1.31 -0.68 5055 1  
## a[2] -0.60 0.00 0.34 -1.15 -0.06 5534 1  
## a[3] -0.24 0.01 0.51 -1.04 0.59 7039 1  
## a[4] -0.19 0.00 0.31 -0.69 0.30 5917 1  
## a[5] -0.57 0.00 0.29 -1.05 -0.11 6277 1  
## a[6] -0.81 0.00 0.25 -1.21 -0.41 6479 1  
## a[7] -0.76 0.00 0.36 -1.35 -0.19 5608 1  
## a[8] -0.52 0.00 0.29 -0.98 -0.04 6466 1  
## a[9] -0.72 0.00 0.35 -1.29 -0.16 5778 1  
## a[10] -1.13 0.01 0.41 -1.79 -0.49 4487 1  
## a[11] -1.55 0.01 0.44 -2.29 -0.90 3667 1  
## a[12] -0.61 0.00 0.31 -1.11 -0.12 5667 1  
## a[13] -0.43 0.00 0.32 -0.95 0.08 6069 1  
## a[14] 0.39 0.00 0.18 0.10 0.69 5843 1  
## a[15] -0.55 0.00 0.33 -1.07 -0.04 5883 1  
## a[16] -0.12 0.00 0.34 -0.65 0.43 5582 1  
## a[17] -0.75 0.00 0.34 -1.28 -0.21 5527 1  
## a[18] -0.64 0.00 0.26 -1.06 -0.24 5578 1  
## a[19] -0.50 0.00 0.31 -1.01 -0.01 5790 1  
## a[20] -0.48 0.00 0.37 -1.04 0.11 5799 1  
## a[21] -0.50 0.00 0.36 -1.07 0.07 6575 1  
## a[22] -0.97 0.01 0.37 -1.58 -0.40 4708 1  
## a[23] -0.77 0.01 0.38 -1.39 -0.18 4982 1  
## a[24] -1.18 0.01 0.43 -1.91 -0.54 3915 1  
## a[25] -0.28 0.00 0.23 -0.64 0.09 6268 1  
## a[26] -0.51 0.01 0.40 -1.16 0.11 6247 1  
## a[27] -1.18 0.00 0.30 -1.66 -0.71 5622 1  
## a[28] -0.96 0.00 0.27 -1.40 -0.54 5642 1  
## a[29] -0.80 0.00 0.31 -1.31 -0.30 5917 1  
## a[30] -0.13 0.00 0.24 -0.51 0.24 6272 1  
## a[31] -0.30 0.00 0.30 -0.77 0.19 5670 1  
## a[32] -0.97 0.01 0.35 -1.55 -0.44 4455 1  
## a[33] -0.42 0.01 0.38 -1.02 0.19 5769 1  
## a[34] 0.28 0.00 0.29 -0.18 0.75 4681 1  
## a[35] -0.13 0.00 0.26 -0.54 0.28 6776 1  
## a[36] -0.58 0.01 0.37 -1.17 0.03 5491 1  
## a[37] -0.24 0.01 0.39 -0.86 0.38 5417 1  
## a[38] -0.70 0.00 0.39 -1.33 -0.10 6436 1  
## a[39] -0.21 0.00 0.32 -0.72 0.30 6005 1  
## a[40] -0.26 0.00 0.26 -0.68 0.15 5896 1  
## a[41] -0.21 0.00 0.33 -0.73 0.32 5572 1  
## a[42] -0.24 0.01 0.41 -0.89 0.41 5222 1  
## a[43] -0.04 0.00 0.25 -0.45 0.38 5471 1  
## a[44] -0.97 0.00 0.35 -1.56 -0.42 5734 1  
## a[45] -0.65 0.00 0.28 -1.10 -0.20 6094 1  
## a[46] 0.00 0.00 0.20 -0.32 0.31 5614 1  
## a[47] -0.34 0.00 0.37 -0.94 0.24 5642 1  
## a[48] -0.08 0.00 0.27 -0.51 0.35 5854 1  
## a[49] -0.87 0.01 0.48 -1.66 -0.13 5388 1  
## a[50] -0.31 0.00 0.36 -0.88 0.27 6148 1  
## a[51] -0.28 0.00 0.29 -0.74 0.19 5618 1  
## a[52] -0.30 0.00 0.24 -0.69 0.08 5434 1  
## a[53] -0.43 0.00 0.36 -1.01 0.13 6851 1  
## a[54] -0.78 0.01 0.45 -1.51 -0.08 5292 1  
## a[55] 0.09 0.00 0.27 -0.33 0.52 4465 1  
## a[56] -1.06 0.00 0.34 -1.61 -0.54 5117 1  
## a[57] -0.30 0.00 0.30 -0.79 0.17 5323 1  
## a[58] -1.00 0.01 0.44 -1.74 -0.33 4223 1  
## a[59] -0.99 0.00 0.32 -1.50 -0.50 5043 1  
## a[60] -1.06 0.00 0.30 -1.55 -0.58 4817 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed Jun 05 17:39:24 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# Draw posterior  
post.8 <- as.data.frame( fit.stan.8 )[,1:60]  
  
p\_mean <- as.data.frame( logistic( apply( post.8 , 2 , mean ) ) )  
colnames(p\_mean) <- c("mu")  
p\_mean$district\_id <- seq( from=1 , to=60 )  
p\_mean <- as.data.frame( p\_mean )  
  
p\_PI <- t( as.data.frame( logistic(apply( post.8 , 2, PI , prob=0.89 ) ) ) )  
district\_id <- seq( from=1 , to=60 )  
p\_PI <- cbind( p\_PI , newColumn=district\_id )  
colnames(p\_PI) <- c("hpdi\_l","hpdi\_h","district\_id")  
p\_PI <- as.data.frame( p\_PI )  
  
# Plot raw data and model estimate of each district  
p <- ggplot()   
p8 <- p +   
 geom\_point(data = data.con.rate,  
 aes(district\_id, rate),   
 shape = 1, color = 'dodgerblue') +  
 scale\_x\_discrete(limits = data.con.rate$x\_label) +   
 theme(axis.text.x = element\_text(angle = 90)) +  
 geom\_point(data = p\_mean,  
 aes(district\_id, mu),   
 shape = 1, color = 'red') +  
 geom\_segment(data = p\_PI,  
 aes(x = district\_id, y = hpdi\_l, xend = district\_id, yend = hpdi\_h), alpha = .5) +  
 ggtitle("Prediction (Multilevel model)")  
  
grid.arrange(p7, p8, nrow = 2)



# Calculate WAIC and compare models  
log\_lik\_7 <- extract\_log\_lik(fit.stan.7, merge\_chains = FALSE)  
waic\_7 <- waic(log\_lik\_7)

## Warning: 4 (0.2%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

log\_lik\_8 <- extract\_log\_lik(fit.stan.8, merge\_chains = FALSE)  
waic\_8 <- waic(log\_lik\_8)  
compare\_model <- loo::compare(waic\_7, waic\_8)  
compare\_model

## elpd\_diff se   
## 9.8 6.6

## Conclusion

By comparing WAIC values, the multilevel model seems to be better. However, from the prediction plots, we can see that in some districts with less data, the model could fit badly, compared to the fixed-effects model.