model2

Lab B Team 2

```r
spa3<-read.csv('spa3.csv')
```

```r
eng3<-read.csv('eng3.csv')
```

```r
#logistic regression model for english data:
M1<-glm(accuracy~cognate, family=binomial(link="logit"),data=eng3)
summary(M1)
```

```
##
## Call:
## glm(formula = accuracy ~ cognate, family = binomial(link = "logit"),
##     data = eng3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0842  -1.0842  -0.9424   1.2735   1.4322
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.58151    0.08555  -6.797 1.07e-11 ***
## cognate      0.35836    0.11890   3.014  0.00258 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1600.5  on 1187  degrees of freedom
## Residual deviance: 1591.4  on 1186  degrees of freedom
## AIC: 1595.4
##
## Number of Fisher Scoring iterations: 4
```

```r
#Intercept and coefficient for cognate:
inv.logit(-0.58)
```

```
## [1] 0.3589326
```

```r
inv.logit(-0.58+0.35)
```

```
## [1] 0.4427521
```

```r
#logistic regression model for spanish data:
M2<-glm(accuracy~cognate, family=binomial(link="logit"),data=spa3)
summary(M2)
```

```
##
## Call:
```

```
## glm(formula = accuracy ~ cognate, family = binomial(link = "logit"),
##     data = spa3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9341  -0.9341  -0.7004   1.4421   1.7466
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.28007    0.09953 -12.861  < 2e-16 ***
## cognate      0.67654    0.13143   5.148 2.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1420.0  on 1186  degrees of freedom
## Residual deviance: 1392.9  on 1185  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1396.9
##
## Number of Fisher Scoring iterations: 4
```

```r
inv.logit(-1.28)
```

```
## [1] 0.2175502
```

```r
inv.logit(-1.28+0.67)
```

```
## [1] 0.3520592
```

Add random variable into logistic model

```r
M3 <- glmer(accuracy~1+(1|cognate),data=eng3,family=binomial(link="logit"))
print(M3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ 1 + (1 | cognate)
##    Data: eng3
##      AIC       BIC    logLik  deviance  df.resid
## 1600.4328 1610.5929 -798.2164 1596.4328      1186
## Random effects:
##  Groups  Name        Std.Dev.
##  cognate (Intercept) 0.1585
## Number of obs: 1188, groups:  cognate, 2
## Fixed Effects:
## (Intercept)
##     -0.4016
```

```r
M4 <- glmer(accuracy~1+(1|cognate),data=spa3,family=binomial(link="logit"))
print(M4)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
```

```
##  Family: binomial  ( logit )
## Formula: accuracy ~ 1 + (1 | cognate)
##    Data: spa3
##       AIC      BIC    logLik  deviance  df.resid
## 1404.1287 1414.2871 -700.0644 1400.1287      1185
## Random effects:
##  Groups  Name        Std.Dev.
##  cognate (Intercept) 0.326
## Number of obs: 1187, groups:  cognate, 2
## Fixed Effects:
## (Intercept)
##       -0.94
```

refit the model with subject levels

```
M5 <- glmer(accuracy~cognate+(1|subject)+diff,data=eng3,family=binomial(link="logit"))
print(summary(M5))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ cognate + (1 | subject) + diff
##    Data: eng3
##
##      AIC      BIC    logLik deviance df.resid
##    947.9    968.2   -469.9    939.9     1184
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9507 -0.3735 -0.0971  0.3559  5.2318
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 6.617    2.572
## Number of obs: 1188, groups:  subject, 27
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.08652    0.56691   3.681 0.000233 ***
## cognate      0.74409    0.17385   4.280 1.87e-05 ***
## diff        -0.88052    0.07006 -12.568  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) cognat
## cognate -0.096
## diff    -0.410 -0.153
```

```
inv.logit(2.08652)
```

```
## [1] 0.8895861
```

```
inv.logit(2.08652+0.74409)
```

```
## [1] 0.9443077
```

```
inv.logit(2.08652-0.8832)
```

```
## [1] 0.7691149
```

Interpret: Intercept: For non-cognate words with same difficulty level, the possibility of answer the words accurately is 88.96%. Cognate: Words with same difficulty level, cognate worlds tend to have 94.43% higher possibility to be correctly answered. diff: For non-Cognates word, it has 76.91% higher possibility to be correctly answered as the difficulty increasing.

```
M6 <- glmer(accuracy~cognate+(1|subject)+diff,data=spa3,family=binomial(link="logit"))
print(summary(M6))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ cognate + (1 | subject) + diff
##    Data: spa3
##
##      AIC      BIC   logLik deviance df.resid
##    965.1    985.4   -478.5    957.1     1183
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0504 -0.4155 -0.1654  0.3529  6.0991
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 4.1      2.025
## Number of obs: 1187, groups:  subject, 27
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.84823    0.46092   1.840   0.0657 .
## cognate      1.16385    0.17533   6.638 3.18e-11 ***
## diff        -0.83898    0.06712 -12.500  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) cognat
## cognate -0.112
## diff    -0.402 -0.256
```

Interpret: Intercept: For non-cognate words with same difficulty level, the possibility of answer the words accurately is 74.17%. Cognate: Words with same difficulty level, cognate worlds tend to have 76.195% higher possibility to be correctly answered. diff: For non-Cognates word, it has 30.2% higher possibility to be correctly answered as the difficulty increasing.

```
inv.logit(0.84823)
```

```
## [1] 0.7001957
```

```
inv.logit(0.84823+1.16385)
```

```
## [1] 0.8820596
```

```
inv.logit(0.84823-0.83898)
```

[1] 0.5023125

Interpret: Intercept: For non-cognate words with same difficulty level, the possibility of answer the words accurately is 70.02%. Cognate: Words with same difficulty level, cognate worlds tend to have 88.20% higher possibility to be correctly answered. diff: For non-Cognates word, it has 50.23% higher possibility to be correctly answered as the difficulty increasing.

Mixed effect logistic regression

```
#comb <- read.csv("comb.csv",header=T)
#comb <- unite(comb,"accuracy",c("eng.acc","spa.acc"),sep="",remove = F)
#comb<- comb[-115,]
#comb$category <- rep(NA,1143)
#for (i in 1:1143){
 # if (comb$eng.acc[i] == 0 && comb$spa.acc[i] == 0) {
  #   comb$category[i] <- 1
  #}
  #if (comb$eng.acc[i] == 0 && comb$spa.acc[i] == 1) {
  #   comb$category[i] <- 2
  #}
 ## if (comb$eng.acc[i] == 1 && comb$spa.acc[i] == 0) {
   # comb$category[i] <- 3
  #}
 # if (comb$eng.acc[i] == 1 && comb$spa.acc[i] == 1) {
 #    comb$category[i] <- 4
 # }
```

```
#multinomial model
#M7 <- polr(factor(category)~cognate+diff,data=comb)
#print(summary(M7))
```

Try Multinomial analysis with brms

Note: One NA in spa3: no accuracy for word "knocker" (ID: BUBA46).

```
M7 <- brm(accuracy~cognate+(1|subject)+diff,data=spa3,family=bernoulli,prior = c(set_prior("normal(0,8)"
```

```
## Warning: Rows containing NAs were excluded from the model.

## Compiling Stan program...

## Start sampling

##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000285 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.85 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.89218 seconds (Warm-up)
## Chain 1:                1.75196 seconds (Sampling)
## Chain 1:                3.64414 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 9.4e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.94 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 1.89694 seconds (Warm-up)
## Chain 2:                1.87945 seconds (Sampling)
## Chain 2:                3.77639 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 9.3e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.93 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
```

```
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 1.86729 seconds (Warm-up)
## Chain 3:                1.87509 seconds (Sampling)
## Chain 3:                3.74239 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000106 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.06 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 1.77223 seconds (Warm-up)
## Chain 4:                1.53699 seconds (Sampling)
## Chain 4:                3.30922 seconds (Total)
## Chain 4:
```

```r
print(summary(M7))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff
##    Data: spa3 (Number of observations: 1187)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##         total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    2.19      0.40     1.53     3.14 1.01      582     1386
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.91      0.51    -0.16     1.86 1.01      420      671
## cognate       1.18      0.18     0.84     1.55 1.00     2713     2427
## diff         -0.84      0.07    -0.98    -0.71 1.00     2864     2574
##
```

```
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

M8 <- brm(accuracy~cognate+(1|subject)+diff,data=eng3,family=bernoulli,prior = c(set_prior("normal(0,8)

## Compiling Stan program...

## recompiling to avoid crashing R session

## Start sampling

##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000138 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.38 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.70153 seconds (Warm-up)
## Chain 1:                1.58818 seconds (Sampling)
## Chain 1:                3.2897 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8.3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.83 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 1.78575 seconds (Warm-up)
## Chain 2:                 1.90972 seconds (Sampling)
## Chain 2:                 3.69547 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 9.7e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.97 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 1.75837 seconds (Warm-up)
## Chain 3:                 1.40329 seconds (Sampling)
## Chain 3:                 3.16167 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 8.4e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.84 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 1.92689 seconds (Warm-up)
## Chain 4:                 1.55427 seconds (Sampling)
```

```
## Chain 4:                     3.48116 seconds (Total)
## Chain 4:

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```
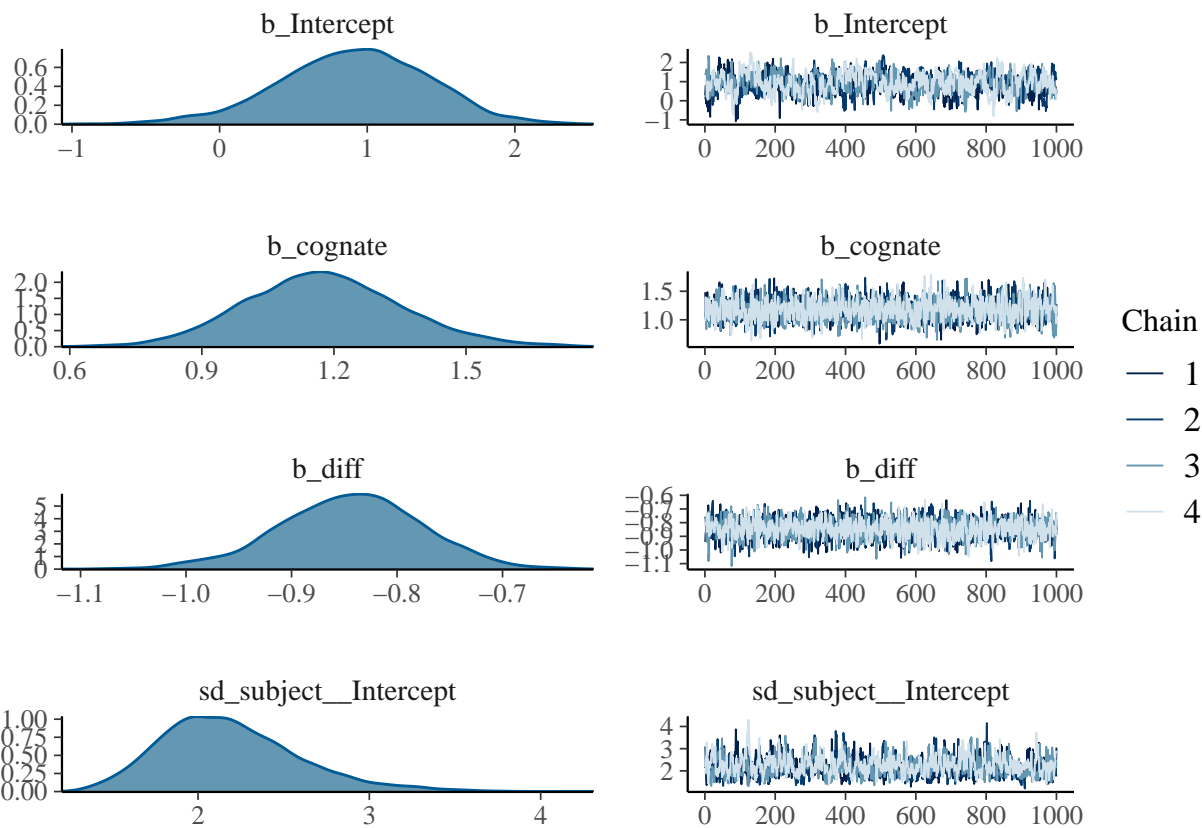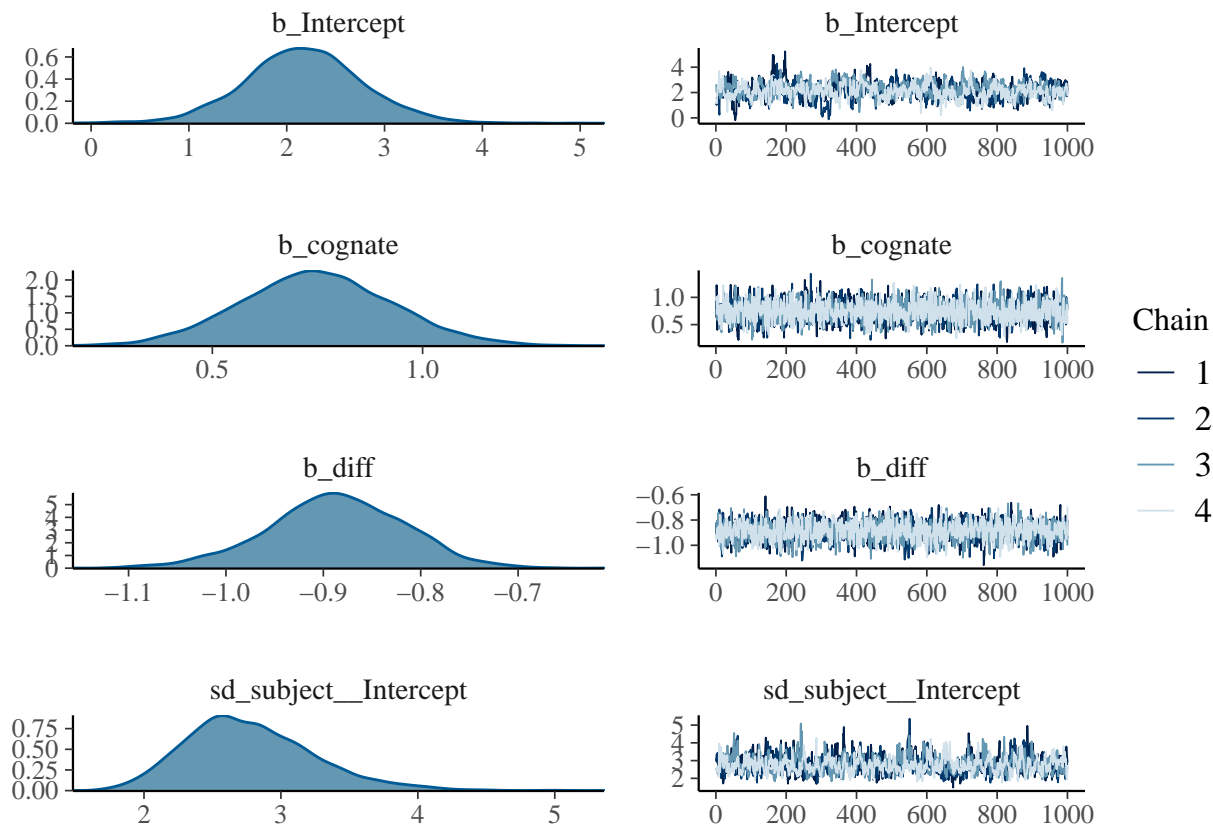
```r
print(summary(M8))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff
##     Data: eng3 (Number of observations: 1188)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.78      0.47     1.99     3.85 1.00      548     1269
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     2.17      0.61     0.99     3.39 1.02      277      589
## cognate       0.75      0.18     0.40     1.10 1.00     2057     2661
## diff         -0.89      0.07    -1.03    -0.75 1.00     2426     2586
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
plot(M7)
```

b_Intercept
b_Intercept

b_cognate
b_cognate

Chain
— 1
— 2
— 3
— 4

b_diff
b_diff

sd_subject__Intercept
sd_subject__Intercept

```
plot(M8)
```

b_Intercept
b_Intercept

b_cognate
b_cognate

Chain
— 1
— 2
— 3
— 4

b_diff
b_diff

sd_subject__Intercept
sd_subject__Intercept

Adding L2AoA predictor.

L2:nonnative language AoA: age of acquisition

```
M9<- brm(accuracy~cognate+(1|subject)+diff+L2AoA,data=spa3,family=bernoulli,prior = c(set_prior("normal
```

```
## Warning: Rows containing NAs were excluded from the model.

## Compiling Stan program...

## recompiling to avoid crashing R session

## Start sampling

##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000234 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.34 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 3.34454 seconds (Warm-up)
## Chain 1:                2.66432 seconds (Sampling)
## Chain 1:                6.00886 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.81 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 3.40739 seconds (Warm-up)
## Chain 2:                2.5091 seconds (Sampling)
## Chain 2:                5.91649 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.8 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.54273 seconds (Warm-up)
## Chain 3:                2.17569 seconds (Sampling)
## Chain 3:                5.71842 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 9e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.9 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 3.64286 seconds (Warm-up)
```

```
## Chain 4:                    2.68399 seconds (Sampling)
## Chain 4:                    6.32685 seconds (Total)
## Chain 4:
```

```r
print(summary(M9))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff + L2AoA
##    Data: spa3 (Number of observations: 1187)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.19      0.41     1.54     3.10 1.00      833     1598
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.06      0.78    -1.42     1.66 1.00      753     1210
## cognate       1.18      0.18     0.85     1.53 1.00     3249     2677
## diff         -0.85      0.07    -0.98    -0.72 1.00     3929     2940
## L2AoA         0.08      0.06    -0.04     0.18 1.01      763     1203
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
M10 <- brm(accuracy~cognate+(1|subject)+diff+L2AoA,data=eng3,family=bernoulli,prior = c(set_prior("norma
```

```
## Compiling Stan program...

## recompiling to avoid crashing R session

## Start sampling

##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000113 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.13 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
```

```
## Chain 1:
## Chain 1:  Elapsed Time: 3.01771 seconds (Warm-up)
## Chain 1:                2.64387 seconds (Sampling)
## Chain 1:                5.66159 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8.7e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.87 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 2.85269 seconds (Warm-up)
## Chain 2:                2.2983 seconds (Sampling)
## Chain 2:                5.15099 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 8.1e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.81 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.29045 seconds (Warm-up)
## Chain 3:                1.94889 seconds (Sampling)
## Chain 3:                5.23934 seconds (Total)
```
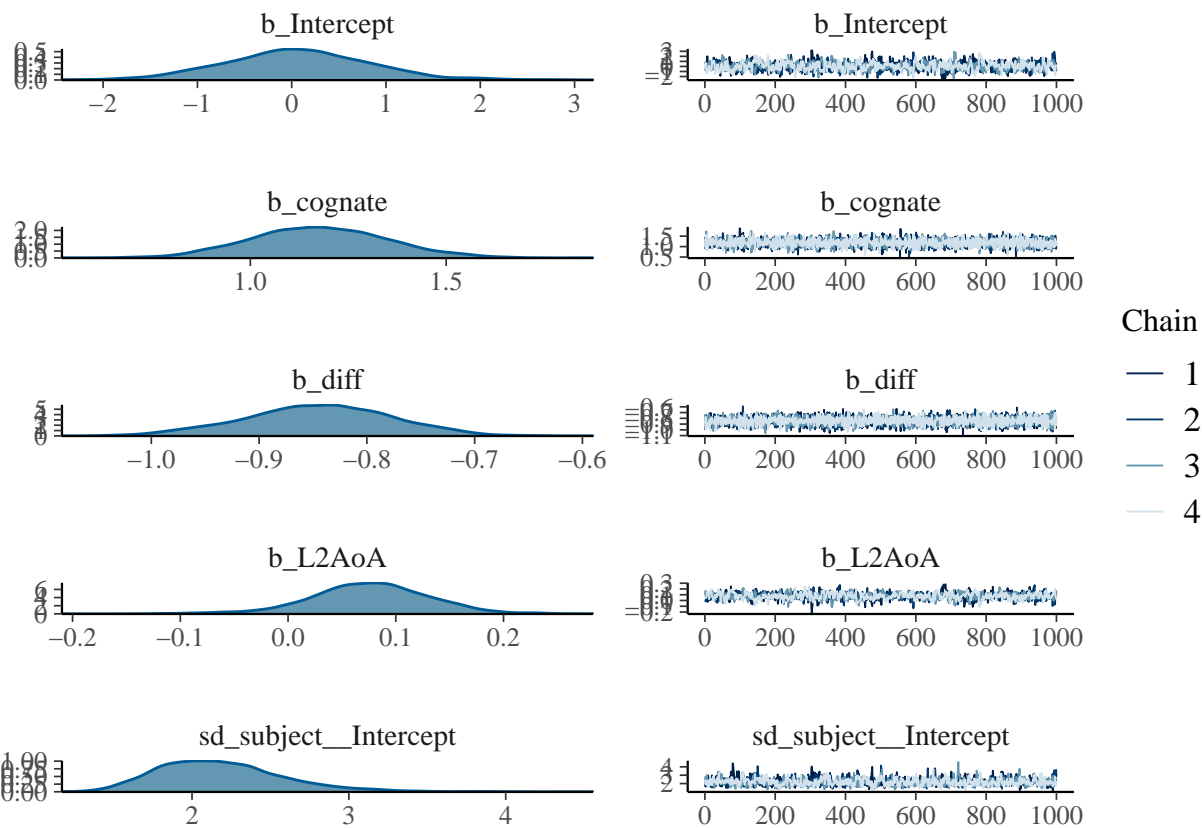
```
## Chain 3:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 8e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.8 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 2.93706 seconds (Warm-up)
## Chain 4:                2.3285 seconds (Sampling)
## Chain 4:                5.26556 seconds (Total)
## Chain 4:
```
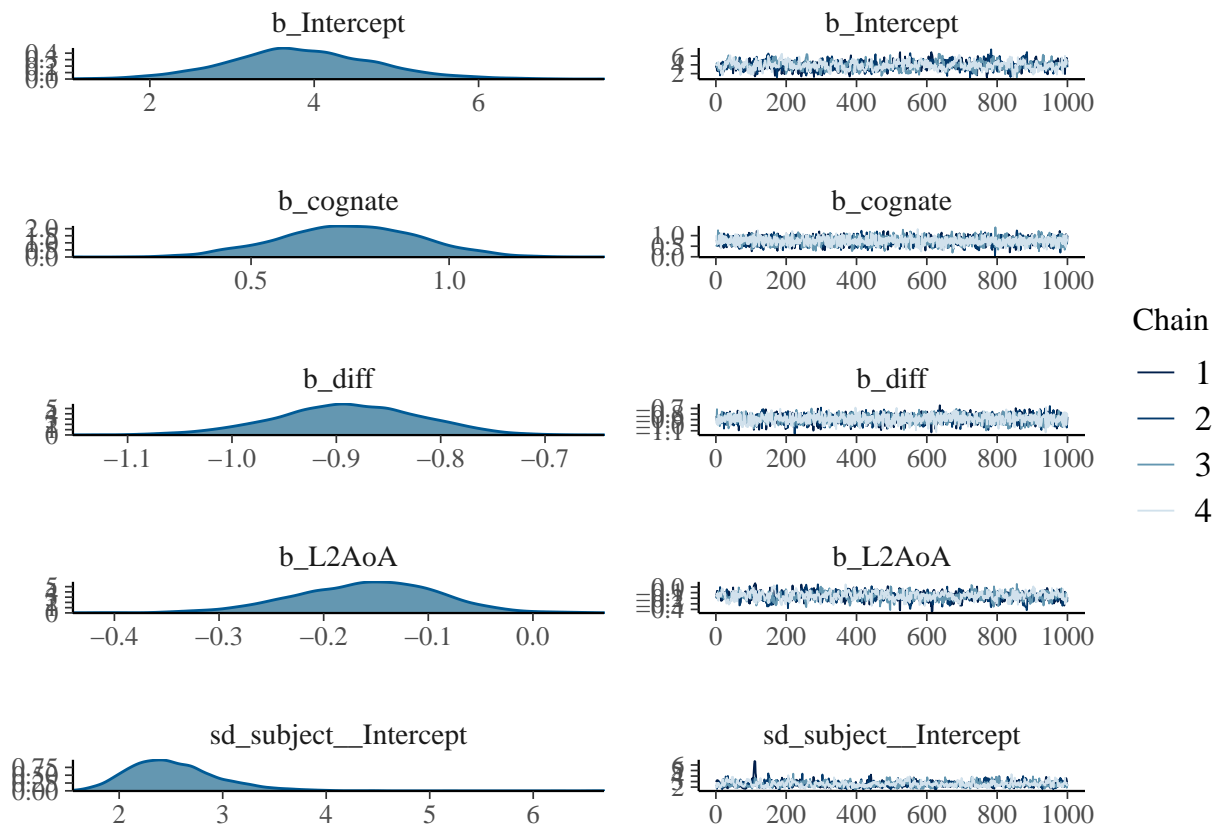
```
print(summary(M10))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff + L2AoA
##    Data: eng3 (Number of observations: 1188)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.53      0.45     1.83     3.53 1.00      742     1199
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     3.85      0.89     2.12     5.70 1.01      637      896
## cognate       0.75      0.17     0.41     1.09 1.00     2915     2794
## diff         -0.89      0.07    -1.03    -0.76 1.00     3136     2608
## L2AoA        -0.16      0.07    -0.30    -0.03 1.01      743      990
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(M9)
```

b_Intercept

b_Intercept

b_cognate

b_cognate

— 1

b_diff

b_diff

— 2

— 3

— 4

b_L2AoA

b_L2AoA

sd_subject__Intercept

sd_subject__Intercept

```
plot(M10)
```

b_Intercept

b_Intercept

b_cognate

b_cognate

Chain

— 1

b_diff

b_diff

— 2

— 3

— 4

b_L2AoA

b_L2AoA

sd_subject__Intercept

sd_subject__Intercept

17

Revision: may need to include language: language=0: spanish as L1 native language; english nonnative language=1: english as L1 native language;spanish nonnative

```
M11<- brm(accuracy~cognate+(1|subject)+diff+L2AoA+Language,data=spa3,family=bernoulli,prior = c(set_pri
```

Warning: Rows containing NAs were excluded from the model.

Compiling Stan program...

recompiling to avoid crashing R session

Start sampling

```
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000119 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.19 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 4.03369 seconds (Warm-up)
## Chain 1:                2.9464 seconds (Sampling)
## Chain 1:                6.98009 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8.3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.83 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 4.14123 seconds (Warm-up)
## Chain 2:                2.89046 seconds (Sampling)
## Chain 2:                7.03168 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 8.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.85 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.87716 seconds (Warm-up)
## Chain 3:                2.41605 seconds (Sampling)
## Chain 3:                6.2932 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000306 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 3.06 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 5.13818 seconds (Warm-up)
## Chain 4:                3.38282 seconds (Sampling)
```

```
## Chain 4:                    8.52099 seconds (Total)
## Chain 4:
```

```r
print(summary(M9))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff + L2AoA
##    Data: spa3 (Number of observations: 1187)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.19      0.41     1.54     3.10 1.00      833     1598
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.06      0.78    -1.42     1.66 1.00      753     1210
## cognate       1.18      0.18     0.85     1.53 1.00     3249     2677
## diff         -0.85      0.07    -0.98    -0.72 1.00     3929     2940
## L2AoA         0.08      0.06    -0.04     0.18 1.01      763     1203
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
M12<- brm(accuracy~cognate+(1|subject)+diff+L2AoA+Language,data=eng3,family=bernoulli,prior = c(set_pri
```

```
## Compiling Stan program...
```

```
## recompiling to avoid crashing R session
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000226 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.26 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
```

```
## Chain 1:  Elapsed Time: 4.3165 seconds (Warm-up)
## Chain 1:                3.10182 seconds (Sampling)
## Chain 1:                7.41832 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 8.4e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.84 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 4.03716 seconds (Warm-up)
## Chain 2:                2.71642 seconds (Sampling)
## Chain 2:                6.75358 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 8.3e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.83 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.83186 seconds (Warm-up)
## Chain 3:                3.3982 seconds (Sampling)
## Chain 3:                7.23006 seconds (Total)
## Chain 3:
```

```
## 
## SAMPLING FOR MODEL 'fecdb0ea2e3aab30d18291906b136a6d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 8.3e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.83 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 4.02184 seconds (Warm-up)
## Chain 4:                3.62519 seconds (Sampling)
## Chain 4:                7.64702 seconds (Total)
## Chain 4:
```

```r
print(summary(M10))
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ cognate + (1 | subject) + diff + L2AoA
##    Data: eng3 (Number of observations: 1188)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.53      0.45     1.83     3.53 1.00      742     1199
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     3.85      0.89     2.12     5.70 1.01      637      896
## cognate       0.75      0.17     0.41     1.09 1.00     2915     2794
## diff         -0.89      0.07    -1.03    -0.76 1.00     3136     2608
## L2AoA        -0.16      0.07    -0.30    -0.03 1.01      743      990
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

#Trace and Density plots

```
plot(M11)
```
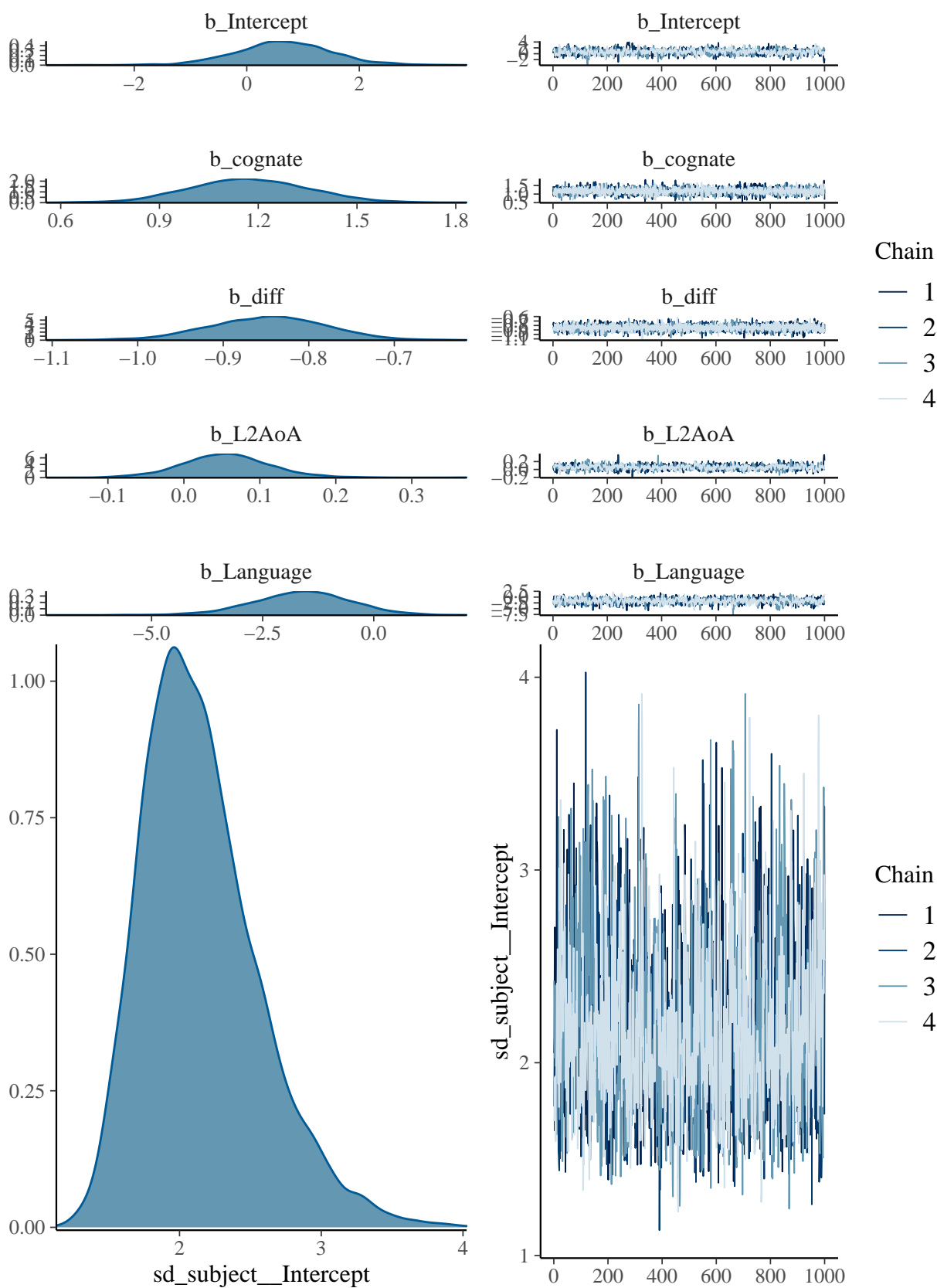
```
plot(M12)
```
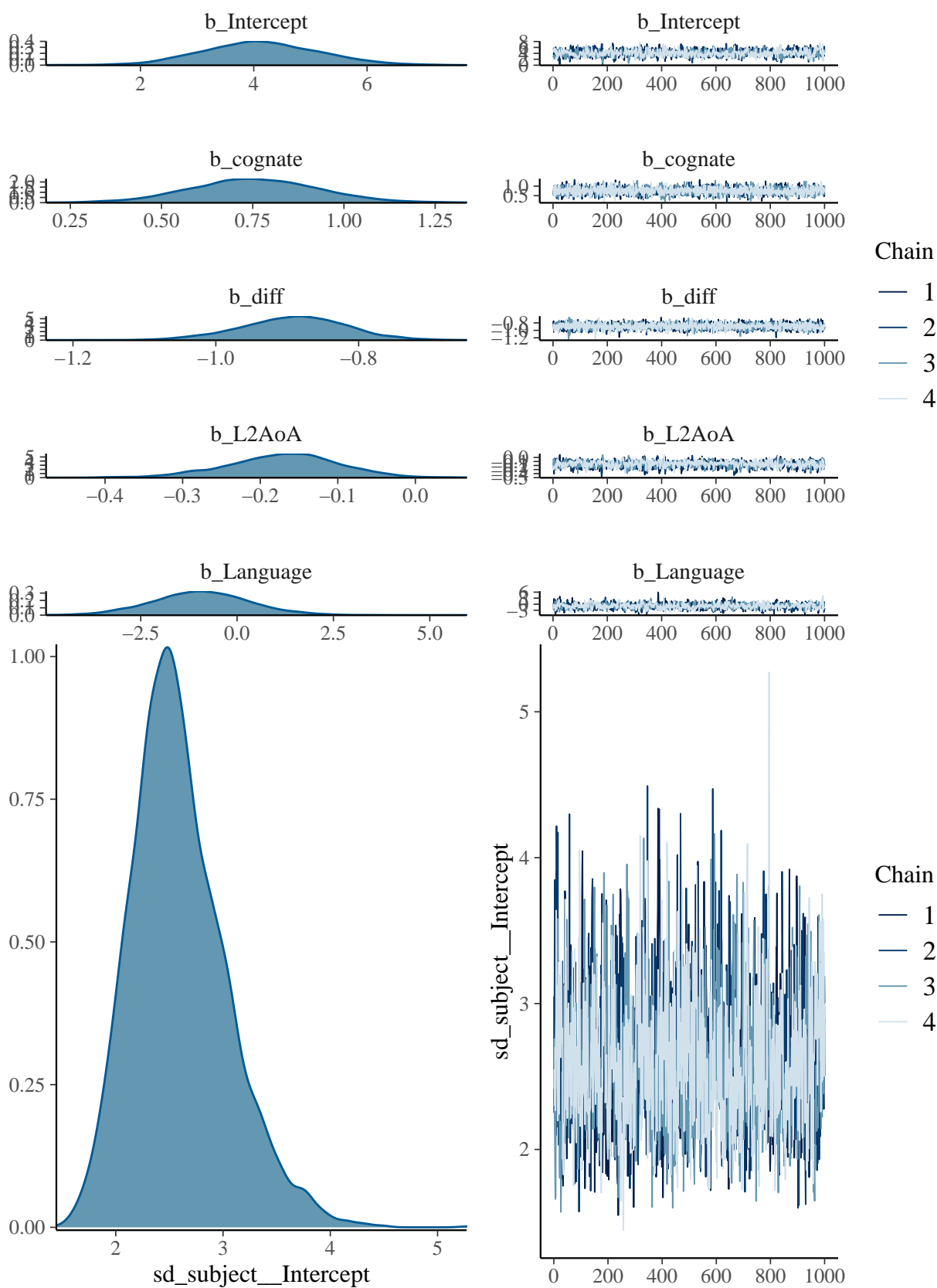
```
waic1<-waic(M7)
```

WAIC and LOO

```
## Warning:
## 2 (0.2%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
waic2<-waic(M8)
```

```
## Warning:
## 2 (0.2%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
waic3<-waic(M11)
```

```
## Warning:
## 2 (0.2%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
waic4<-waic(M12)
```

```
## Warning:
## 2 (0.2%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
loo1<-loo(M7)
loo2<-loo(M8)
```

```
loo3<-loo(M9)
loo4<-loo(M10)
loo5<-loo(M11)
```

```
## Warning: Found 1 observations with a pareto_k > 0.7 in model 'M11'. It is
## recommended to set 'moment_match = TRUE' in order to perform moment matching for
## problematic observations.
```

```
loo6<-loo(M12)
```

```
#Spanish:
loo_compare(loo1,loo3,loo5)
```

```
##     elpd_diff se_diff
## M7   0.0       0.0
## M9   0.0       0.4
## M11 -0.4       0.6
```

```
#English:
loo_compare(loo2,loo4,loo6)
```

```
##     elpd_diff se_diff
## M8   0.0       0.0
## M10  0.0       0.3
## M12 -0.1       0.4
```