

# Bayesian inference for How do People Type on Mobile Devices?

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Data example</b>	<b>1</b>
<b>3</b>	<b>Model fitting</b>	<b>3</b>
3.1	WPM prediction . . . . .	3
3.2	Error rate prediction . . . . .	16
	<b>References</b>	<b>28</b>

## 1 Introduction

We use the following libraries to model the problem. BRMS is used for the generalized hierarchical modeling and the linear regression model using Stan.

```
library(rstan)
library(lmerTest)
library(brms)
library("bayesplot")
library("ggplot2")
library(loo)
```

The notebook starts with an introduction about the data set and how to clean the data set. According to this data set, we aim to predict two variables. Hence, we divide the model fitting section into two parts (WPM prediction and Error rate prediction). Moreover, two models are suggested in each part. Last but not least, several plots and convergence measurements are reported in the related sections.

## 2 Data example

We analyze a typing dataset (Palin et al. 2019) which is the observations from a study with 37,000 volunteers on how people type in mobile devices. Data were collected by a web-based transcription task. The task was to transcribe 15 English sentences and answer a questionnaire after. The participants of the task came from a public website ([www.typingtest.com](http://www.typingtest.com)) for training and testing of typing skills. According to the authors of original paper, they collected typing test data from over 260,000 participants. Then, they filtered the participants who did not finish typing 15 sentences and the questionnaires. Also, they further remove about 25% of participants who did not use a mobile device, which yielded a dataset of 37,370 participants typing 15 sentences each. In our case, we got a simplified dataset. We filtered the dataset according to the suggested criterias on the paper. This gives us a dataset containing participants with columns PARTICIPANT\_ID, KEYBOARD\_TYPE, AGE, GENDER, WPM, ERROR\_RATE, USING\_APP, USING\_FEATURES and FINGERS. Notice that the column X below is the indices inherited from the original

dataset, which clearly should be discarded. The simplified dataset only contains the records with mobile as the KEYBOARD\_TYPE. Hence, this columns also should be omitted. We give a summary of the dataset below.

```
# Read data and show summary
```

```
mobile_participants <- read.csv("Data/mobile_participants.csv")
summary(mobile_participants)
```

```
##           X           PARTICIPANT_ID  KEYBOARD_TYPE      AGE
##  Min.      :    0   Min.      :    1   mobile:39149   Min.      :    0.00
##  1st Qu.: 91468   1st Qu.: 91469                1st Qu.:   19.00
##  Median :152808   Median :152809                Median :   23.00
##  Mean   :151344   Mean    :151345                Mean   :   24.71
##  3rd Qu.:217954   3rd Qu.:217955                3rd Qu.:   28.00
##  Max.    :274127   Max.     :274128                Max.    :  3015.00
##
##           GENDER           WPM           ERROR_RATE      USING_APP
##  female:24907   Min.      : -1241.38   Min.      :    0.000   false    :30534
##  male  :12266   1st Qu.:    26.66   1st Qu.:    1.015   notsure  : 5444
##  none   : 1976   Median :    34.52   Median :    1.872   Gboard   : 1026
##                               Mean   :    58.12   Mean     :    3.735   SwiftKey:  834
##                               3rd Qu.:    44.22   3rd Qu.:    3.520   Touchpal:  335
##                               Max.    :168000.00   Max.     :100.000   Go        :  185
##                               (Other) :   791
##
##                               USING_FEATURES
##  ["no"]                      :17302
##  ["autocorrection"]          : 8033
##  ["prediction"]              : 3889
##  ["prediction","autocorrection"] : 3232
##  ["notsure"]                 : 1915
##  ["swipe"]                   : 1082
##  (Other)                     : 3696
##
##                               FINGERS
##  both_hands;thumbs           :30550
##  right_hand;thumbs           : 3641
##  right_hand;index_fingers    : 1115
##  both_hands;index_fingers    :  616
##  left_hand;thumbs            :  543
##  both_hands;other-nfingers;  :  417
##  (Other)                     : 2267
```

Columns AGE, GENDER, USING\_FEATURES, FINGERS show the background information and typing habits of the participants. The median and mean values of the participants ages are 23 and 24.71 respectively. It shows that most participants are young people. We notice that there are also some abnormal values in the column. For example, the participants ages can not be 3015 or 0. We will filter the abnormal values later. In terms of gender ratio, the number of female participants are twice of male participants. USING\_FEATURES shows what kind of input features people used. There are mainly three kinds of features here, autocorrection, prediction and swipe. Notice that the features in this column cannot reflect the 100% truth of what feature each participant used. Authors of the original paper developed an algorithm to recognize the text entry method due to the limitation of the web-based logging in the mobile devices. WPM and ERROR\_RATE are two indicators of participants' input performances, which are also the response variables in this project. WPM represents Word per minute. It is computed as the length of input (one word defined as five characters) divided by the time between the first and the last keystroke. ERROR\_RATE is calculated as

the Levenshtein edit distance between the presented and transcribed string, divided by the larger size of the strings. In this project, we only need AGE, GENDER, WPM, ERROR\_RATE, USING\_FEATURES columns to construct our model and we aim to predict WPM and ERROR\_RATE through them:

```
New_data=mobile_participants[ 1:5000,c("AGE", "GENDER", "USING_FEATURES", "WPM", "ERROR_RATE")
, drop=FALSE]
```

As mentioned before, some preliminary steps are required to clean the data set. These criteria are selected from (Palin et al. 2019). For example, they suggested to exclude some participants which their WPM is more than 200, whose their age is less than 5 or more 61 years old (more than 2 SD away from mean age).

```
Final=subset(New_data, AGE > 5 & AGE < 61 & WPM<200 & WPM>0 & GENDER!="none", ERROR_RATE>0)
```

```
# Filtering USING_FEATURES according to 4 levels (Palin et al. 2019)
levels(Final$USING_FEATURES)[levels(Final$USING_FEATURES)=="[\\no\\"]"] <- "No ITE"
levels(Final$USING_FEATURES)[levels(Final$USING_FEATURES)=="[]"] <- "No ITE"
levels(Final$USING_FEATURES)[levels(Final$USING_FEATURES)
=="[\\prediction\\"]"] <- "prediction"
levels(Final$USING_FEATURES)[levels(Final$USING_FEATURES)
=="[\\autocorrection\\"]"] <- "autocorrection"
levels(Final$USING_FEATURES)[levels(Final$USING_FEATURES)!="No ITE"
& levels(Final$USING_FEATURES)!= "prediction"
& levels(Final$USING_FEATURES)!= "autocorrection"] <- "Mixed ITE"
str(Final)
```

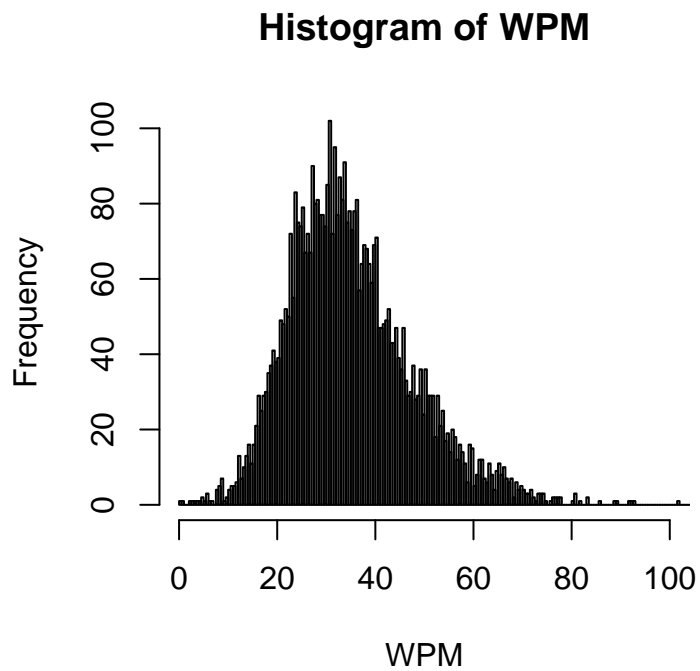
```
## 'data.frame': 4720 obs. of 5 variables:
## $ AGE : int 23 12 22 31 11 49 23 23 22 27 ...
## $ GENDER : Factor w/ 3 levels "female","male",...: 1 1 1 1 1 1 2 2 2 1 ...
## $ USING_FEATURES: Factor w/ 4 levels "Mixed ITE","autocorrection",...: 1 1 3 1 1 1 1 1 1 1 ...
## $ WPM : num 40.5 25.2 21.5 88.8 13.3 ...
## $ ERROR_RATE : num 6.393 2.525 0.575 0.468 2.155 ...
```

## 3 Model fitting

### 3.1 WPM prediction

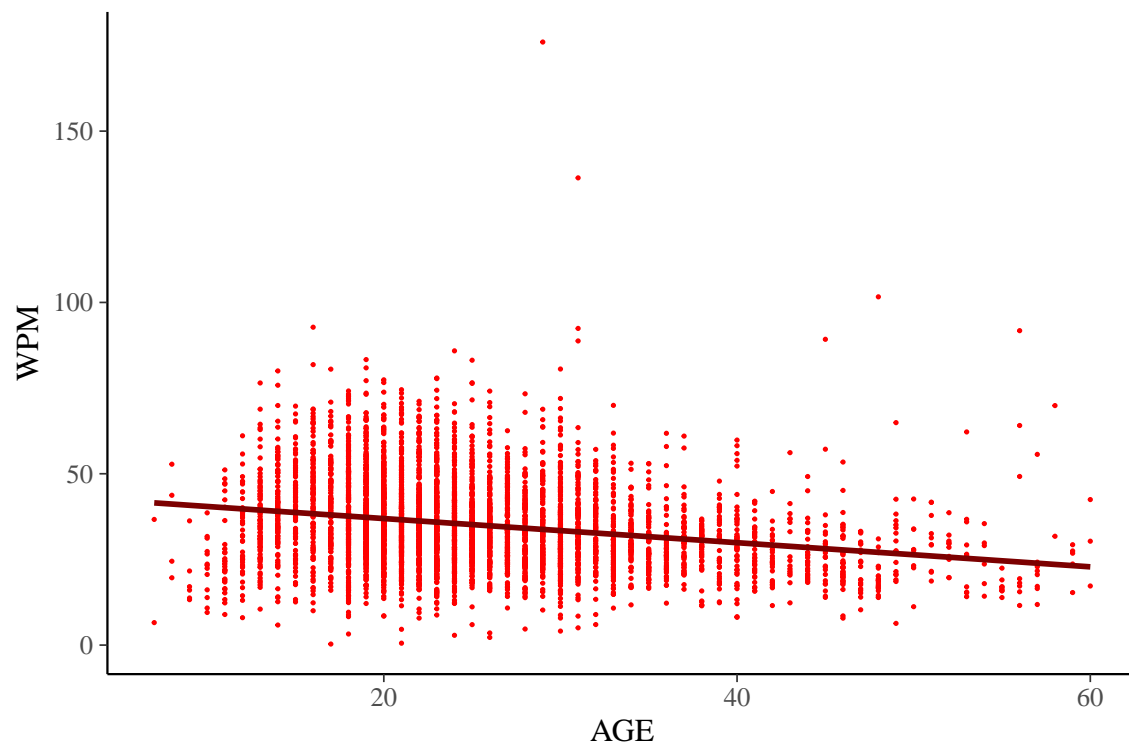
This plot shows the histogram of WPM.

```
hist(Final$WPM, nclass=500,xlim=c(0,100),xlab = "WPM",main="Histogram of WPM")
```



This plot shows the trend of WPM varies by AGE.

```
theme_set(bayesplot::theme_default())
p=ggplot(Final,
  aes(
    x=AGE,
    y=WPM
  ))
p1=p+
  geom_point(size=.25, color = "red")+
  geom_smooth(method="lm",se=FALSE, color="#7c0000")
p1
```



### 3.1.1 Linear regression model

In the first model, we aim to fit the data via a linear regression model. In this model, we only have one input, which is AGE, and the model attempts to find the response, which is WPM. This model can be formulated as:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

#### 3.1.1.1 Stan code

This is the Stan code for our linear regression model. It defines the problem in Stan syntax. In the “generated quantities” block, we draw from posterior predictive distribution to let us visualize how a predictor affects.

```
smodel_wpm_l="data {
  int < lower = 1 > N; // Sample size
  vector[N] x; // Predictor
  vector[N] y; // Outcome
}

parameters {
  real alpha; // Intercept
  real beta; // Slope (regression coefficients)
  real < lower = 0 > sigma; // Error SD
}

model {
  y ~ normal(x * beta + alpha, sigma);
}
```

```

    alpha ~ normal(40, 20);
    beta ~ normal(1, 0.1);
  }

generated quantities {
  real y_rep_wpm[N];
  vector[N] log_lik;
  for (n in 1:N) {
    y_rep_wpm[n] = normal_rng(x[n] * beta + alpha, sigma);
    log_lik[n] = normal_lpdf(y[n] | x[n] * beta + alpha, sigma);
  }
}
"
stan_wpm_1 <- stan_model(model_code = smodel_wpm_1)

data_wpm_1 <- list (x = Final$AGE,y=Final$WPM,N = length(Final$WPM))
fit_wpm_1 <- sampling(stan_wpm_1, data=data_wpm_1, control = list(max_treedepth = 15))

```

### 3.1.1.2 Prior distributions

We use informative priors for  $\alpha \sim \text{normal}(40, 20)$  and  $\beta \sim \text{normal}(1, 0.1)$ . As shown in the trend plot in section WPM prediction, the average WPM is around 40. We set the sd to 20 to pick some sparse priors as well. Moreover, mean 1 and sd 0.1 were selected for the slope prior.

### 3.1.1.3 R\_hat

$\hat{R}$  is a measurement for convergence analysis and the value below 1.1 shows stability of outcomes As  $\hat{R}$  is below 1.1, the convergence is satisfactory. Hence, the output of our model is converged.

```
print(fit_wpm_1, pars=c("alpha", "beta","sigma"), probs=c(.1,.5,.9))
```

```

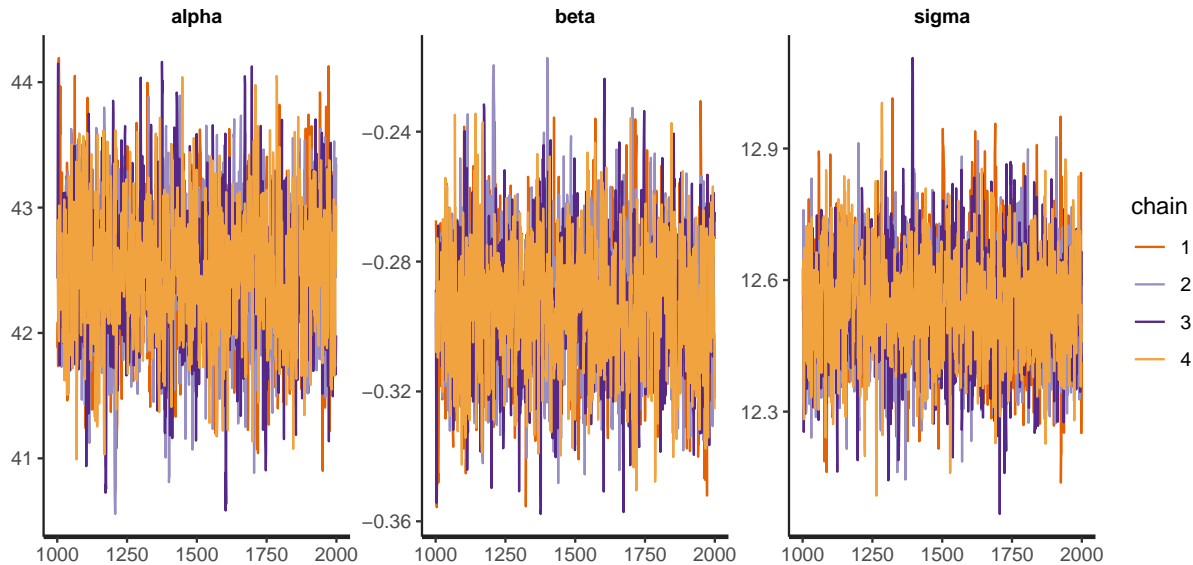
## Inference for Stan model: 7b683b731366d194ea3d8caf27b2bf43.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean   sd  10%  50%  90% n_eff Rhat
## alpha 42.48    0.01 0.56 41.77 42.49 43.20 1635    1
## beta  -0.29    0.00 0.02 -0.32 -0.29 -0.27 1653    1
## sigma 12.54    0.00 0.13 12.37 12.54 12.70 2249    1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 20:12:56 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).

```

### 3.1.1.4 hmc convergence divergences

A useful diagnostic plot is the trace plot. It shows a time series of the Markov chains. These plots show our model is converged.

```
traceplot(fit_wpm_1, pars =c("alpha", "beta","sigma"))
```



### 3.1.1.5 ESS

The effective sample size (ESS) is a quantity measurement showing how many independent draws contain the same amount of information. Higher value of ESS is better. If the value of this parameter is small, then it shows the posterior distribution will be poor.

```
print(fit_wpm_1, pars=c("alpha", "beta", "sigma"), probs=c(.1, .5, .9))
```

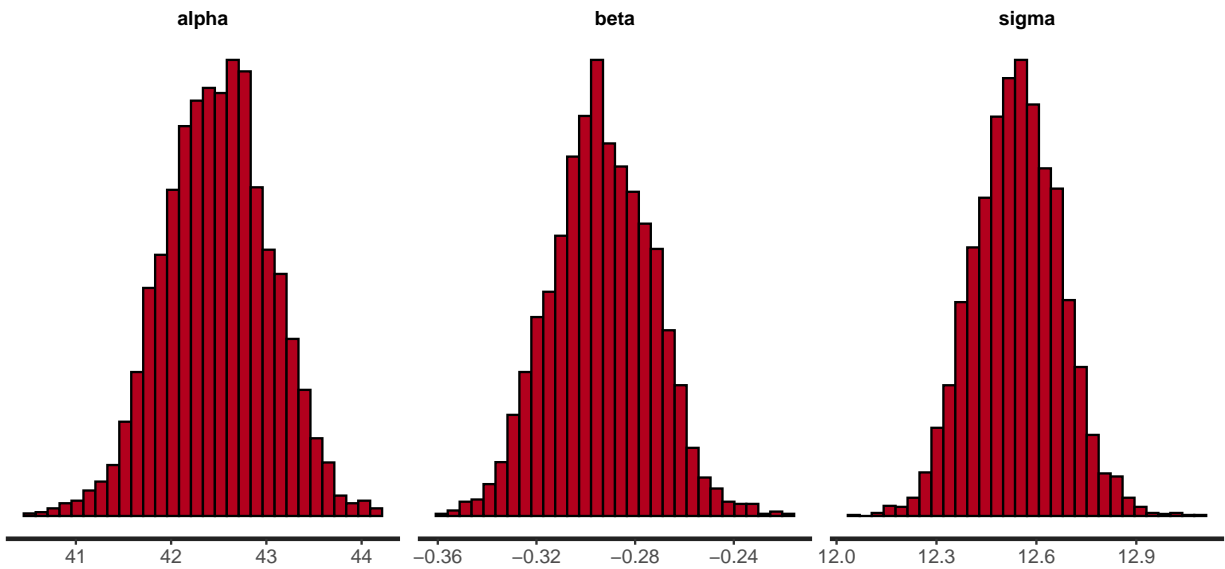
```
## Inference for Stan model: 7b683b731366d194ea3d8caf27b2bf43.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean  sd  10%  50%  90% n_eff Rhat
## alpha 42.48    0.01 0.56 41.77 42.49 43.20 1635    1
## beta -0.29    0.00 0.02 -0.32 -0.29 -0.27 1653    1
## sigma 12.54    0.00 0.13 12.37 12.54 12.70 2249    1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 20:12:56 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).
```

The ESS (it shows under column `n_eff`) values for our model show that it is good.

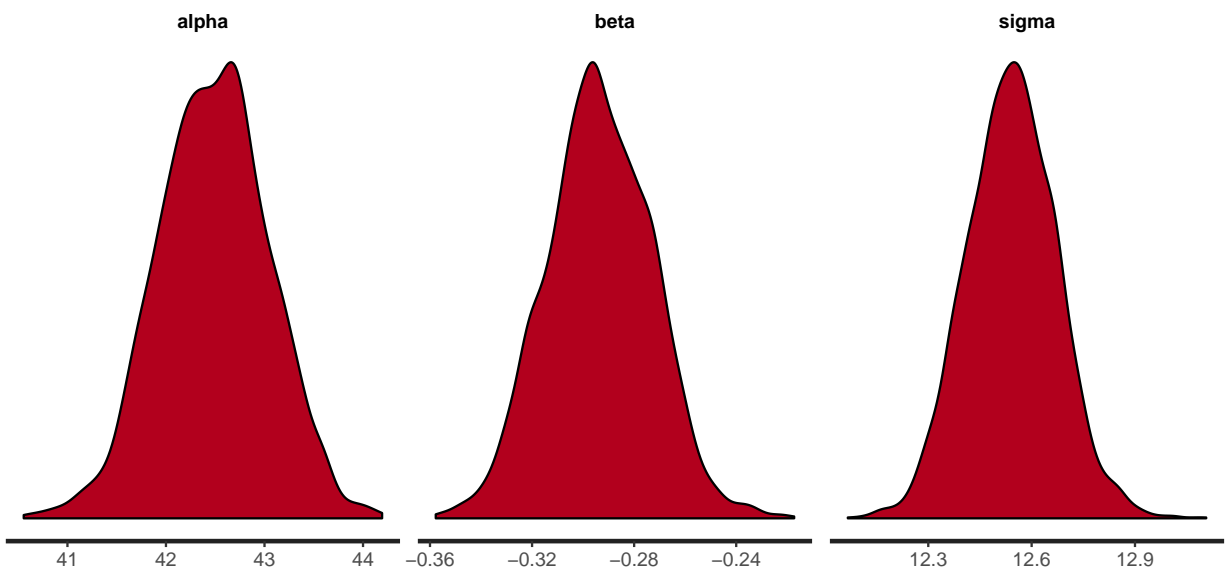
### 3.1.1.6 Posterior check

We can also look at the posterior densities & histograms

```
stan_hist(fit_wpm_1, pars=c("alpha", "beta", "sigma"))
```

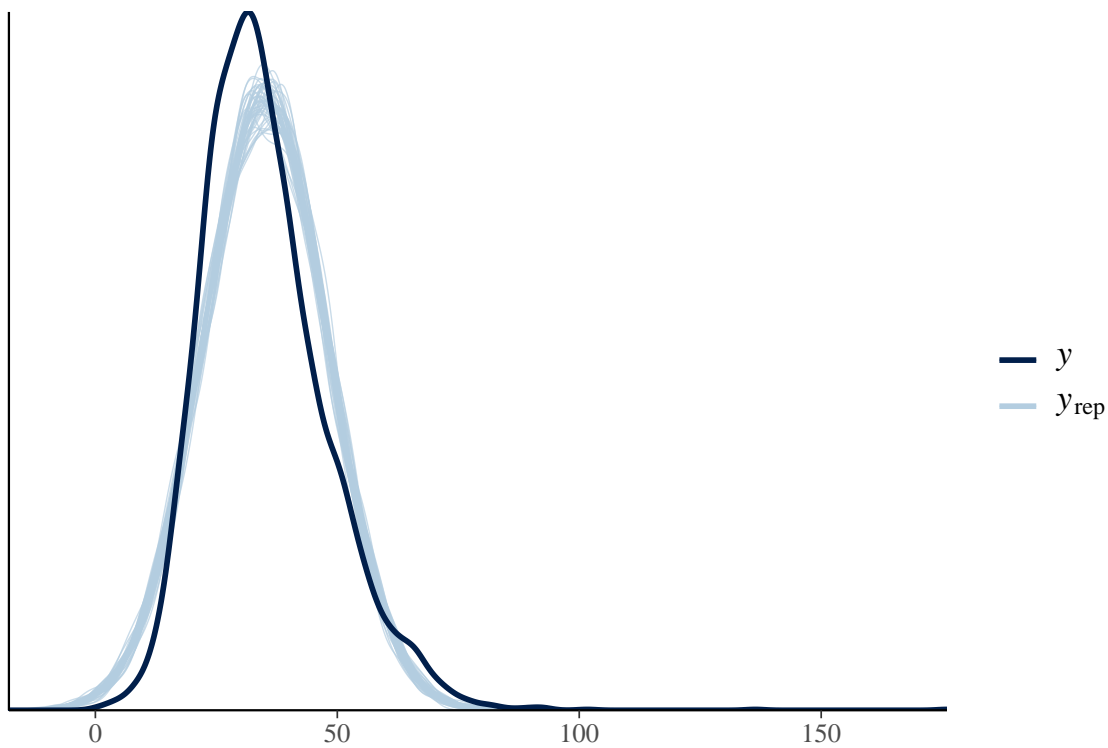


```
stan_dens(fit_wpm_l, pars=c("alpha", "beta", "sigma"))
```

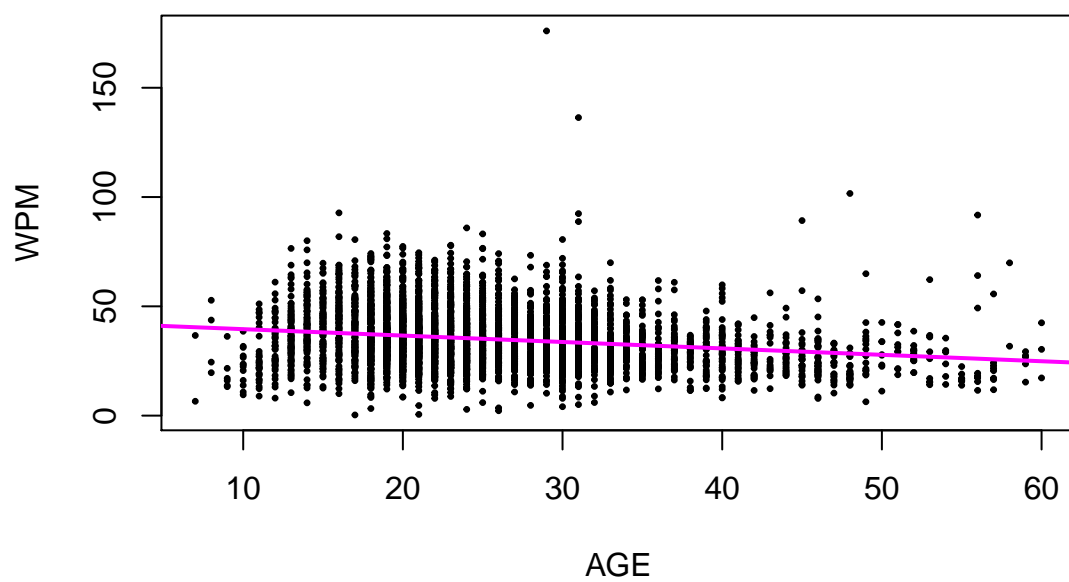


```
y_rep_wpm <- as.matrix(fit_wpm_l, pars = "y_rep_wpm")
ppc_dens_overlay(Final$WPM, y_rep_wpm[1:50, ])
```

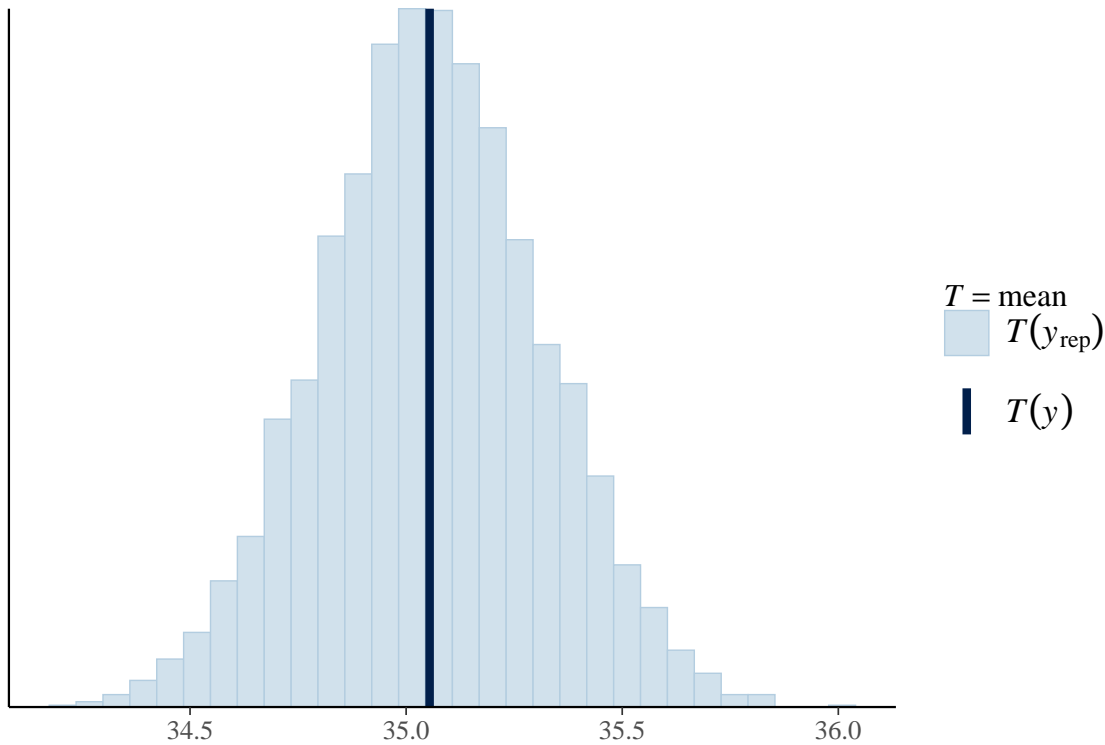




```
posterior <- extract(fit_wpm_1)
plot(Final$WPM ~ Final$AGE, pch = 20, cex = .5, xlab="AGE", ylab="WPM", main="")
abline( mean(posterior$alpha), mean(posterior$beta), col = 6, lw = 2)
```



```
ppc_stat(y = Final$WPM, yrep = y_rep_wpm, stat = "mean")
```



These posterior plots show the model fits to the model and it is able to predict WPM.

### 3.1.2 Hierarchical generalized linear model

Now, we use a hierarchical generalized linear model to fit the model. This model extend linear models to include coefficients that vary by discrete groups.

$$y_i \sim \text{Normal}(\alpha_{j[i]} + x_i' \beta_{j[i]}, \sigma_{j[i]})$$

In this model, parameters are nested within one another.

#### 3.1.2.1 brms code

For this part, we used brms which provides an interface to fit our hierarchical model using Stan.

```
prior <- c(prior_string("normal(40,10)", class = "b"))
fit_wpm_h <- brm(WPM ~ 1+AGE+GENDER+(1+GENDER|USING_FEATURES), data = Final,
                 control = list(max_treedepth = 15), prior = prior)
fit_wpm_h <- add_criterion(fit_wpm_h, "waic")
```

#### 3.1.2.2 Prior distributions

We use informative priors for  $\beta \sim \text{normal}(40, 10)$  population-level effects. As shown in the trend plot in section WPM prediction, the WPM is around 40. We set the sd to 10 to pick some sparse priors as well.

### 3.1.2.3 R\_hat

$\hat{R}$  is a measurement for convergence analysis and the value below 1.1 shows stability of outcomes. As  $\hat{R}$  is below 1.1, the convergence is satisfactory. Hence, the output of our model is converged.

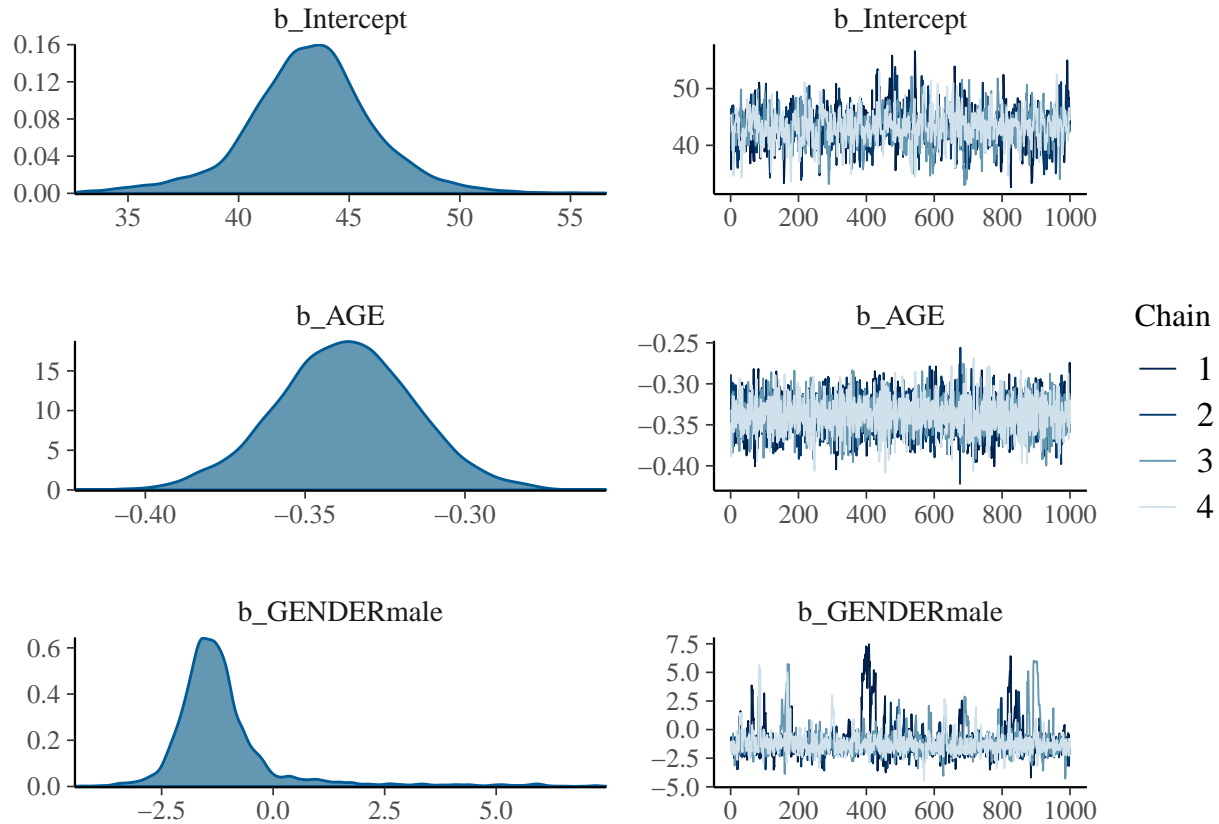
```
print(fit_wpm_h, pars=c("alpha", "beta", "sigma"), probs=c(.1,.5,.9))

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: WPM ~ 1 + AGE + GENDER + (1 + GENDER | USING_FEATURES)
## Data: Final (Number of observations: 4720)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~USING_FEATURES (Number of levels: 4)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat
## sd(Intercept)      5.63      2.93   2.17   13.41 1.00
## sd(GENDERmale)     1.24      1.81   0.03    6.69 1.01
## cor(Intercept,GENDERmale) -0.07    0.59  -0.97    0.94 1.00
##           Bulk_ESS Tail_ESS
## sd(Intercept)      1137    1649
## sd(GENDERmale)      567     315
## cor(Intercept,GENDERmale) 2302    2239
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      43.23      2.90   36.87   49.15 1.00     965    1216
## AGE            -0.34      0.02   -0.38   -0.30 1.00    3966    2105
## GENDERmale     -1.13      1.27   -2.65    2.80 1.01     751     266
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      12.30      0.13   12.05   12.54 1.00    4293    3196
##
## 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).
```

### 3.1.2.4 hmc convergence divergences

A useful diagnostic plot is the trace plot. It shows a time series of the Markov chains. These plots show our model is converged.

```
plot(fit_wpm_h, pars = "~b_")
```



### 3.1.2.5 ESS

The effective sample size (ESS) is a quantity measurement showing how many independent draws contain the same amount of information. Higher value of ESS is better. If the value of this parameter is small, then it shows the posterior distribution will be poor.

```
print(fit_wpm_h, pars=c("alpha", "beta", "sigma"), probs=c(.1, .5, .9))
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: WPM ~ 1 + AGE + GENDER + (1 + GENDER | USING_FEATURES)
## Data: Final (Number of observations: 4720)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~USING_FEATURES (Number of levels: 4)
##
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat
sd(Intercept)	5.63	2.93	2.17	13.41	1.00
sd(GENDERmale)	1.24	1.81	0.03	6.69	1.01
cor(Intercept, GENDERmale)	-0.07	0.59	-0.97	0.94	1.00

```
## Bulk_ESS Tail_ESS
## sd(Intercept)      1137    1649
## sd(GENDERmale)      567     315
## cor(Intercept, GENDERmale) 2302    2239
```

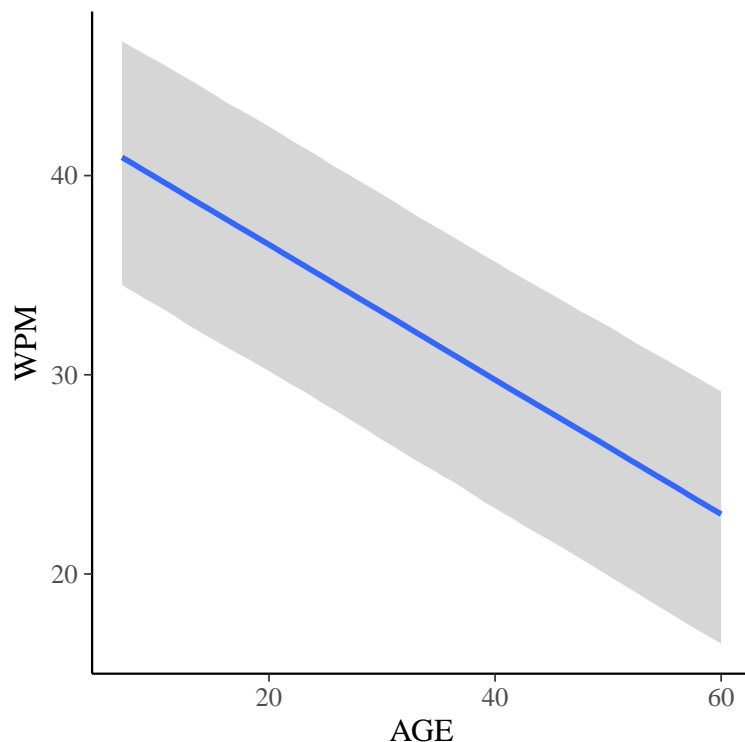
```
##
## Population-Level Effects:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    43.23     2.90   36.87   49.15 1.00     965    1216
## AGE         -0.34     0.02   -0.38   -0.30 1.00    3966    2105
## GENDERmale   -1.13     1.27   -2.65    2.80 1.01     751     266
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.30     0.13   12.05   12.54 1.00    4293    3196
##
## 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).
```

The ESS values for our model show that it is good.

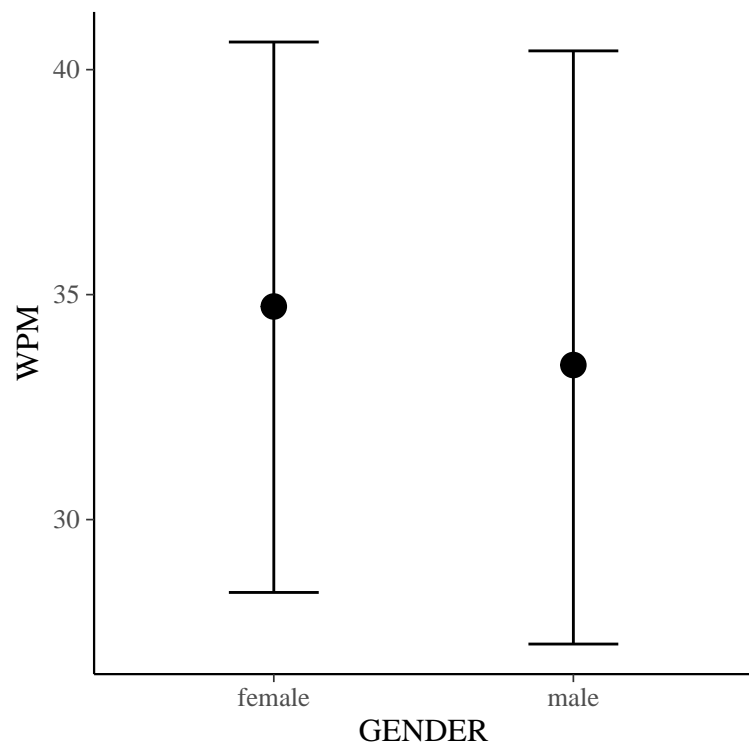
### 3.1.2.6 Postrior check

We can also look at the posterior densities & histograms

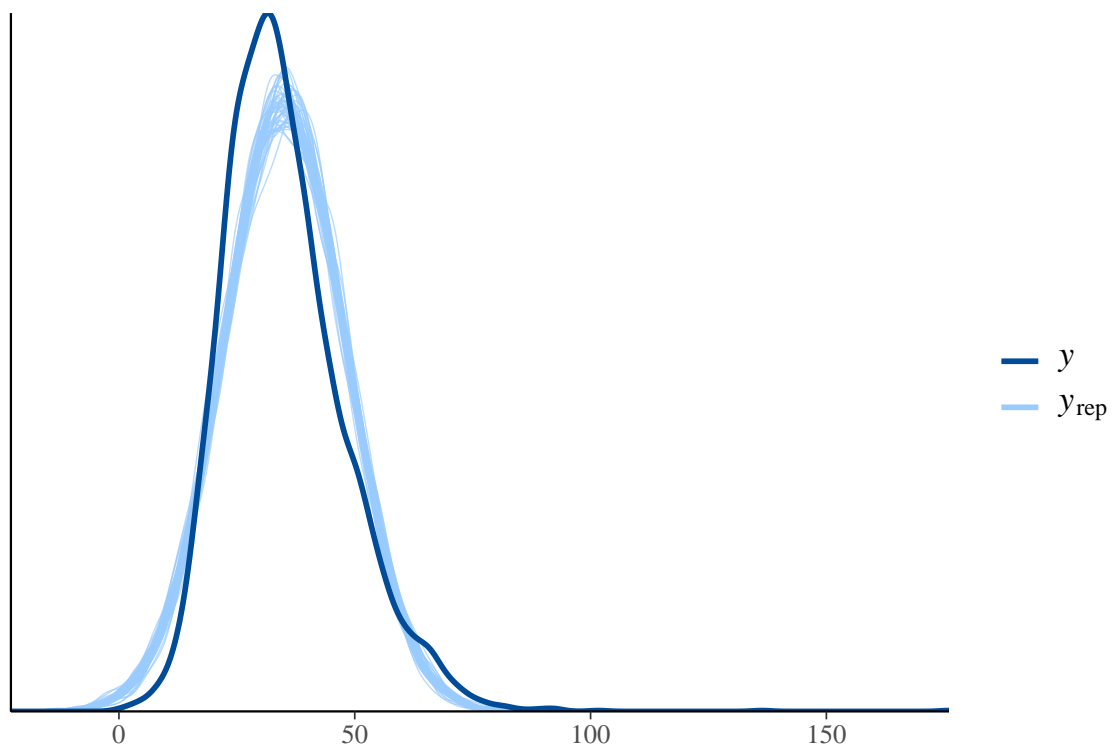
```
marginal_effects(fit_wpm_h, effects = "AGE")
```



```
marginal_effects(fit_wpm_h, effects = "GENDER")
```

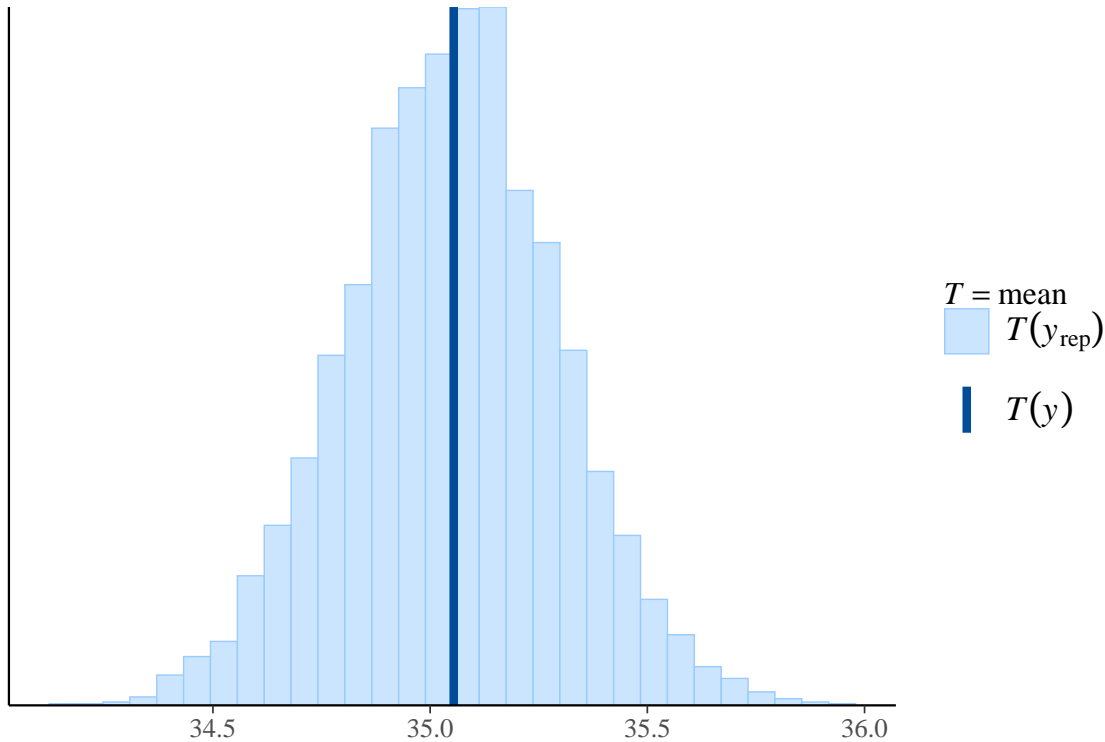


```
yrep_wpm_h <- posterior_predict(fit_wpm_h, draws = 500)
color_scheme_set("brightblue")
ppc_dens_overlay(Final$WPM, yrep_wpm_h[1:50, ])
```



```
ppc_stat(y = Final$WPM, yrep = yrep_wpm_h, stat = "mean")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



These posterior plots show the model fits to the model and it is able to predict WPM.

### 3.1.2.7 Comparison and conclusion

We used loo to compare the models. It is applied to estimate the difference in the expected predictive accuracy.

```
loo_wpm_1 <- loo(fit_wpm_1, save_psis = TRUE)
print(loo_wpm_1)
```

```
##
## Computed from 4000 by 4720 log-likelihood matrix
##
##      Estimate      SE
## elpd_loo -18635.7  93.9
## p_loo      5.9    1.9
## looic     37271.5 187.7
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_wpm_h <- loo(fit_wpm_h, save_psis = TRUE)
print(loo_wpm_h)

##
## Computed from 4000 by 4720 log-likelihood matrix
##
##      Estimate      SE
## elpd_loo -18548.1  95.3
## p_loo      11.4   2.0
## looic      37096.2 190.6
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
compare(loo_wpm_l, loo_wpm_h)
```

```
## elpd_diff      se
##      87.7      13.3
```

The difference in ELPD is much larger than twice the estimated standard error again indicating that the hierarchical model is expected to have better predictive performance than the linear model.

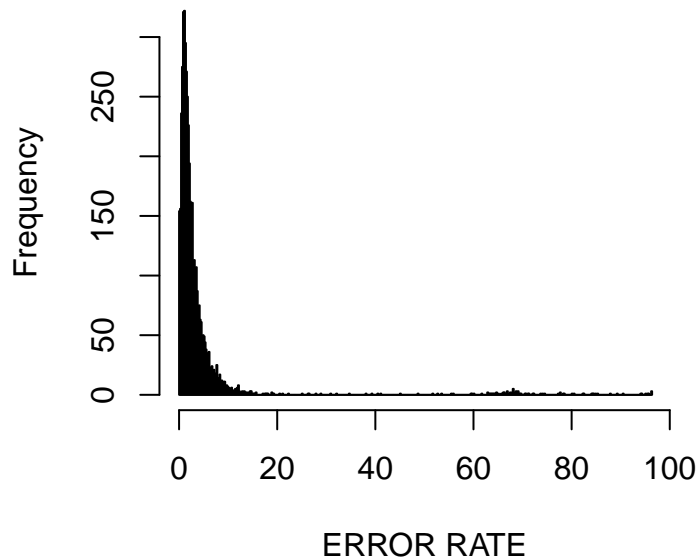
### 3.2 Error rate prediction

This plot shows the histogram of the error rate.

```
hist(Final$ERROR_RATE, nclass=500,xlim=c(0,100),col="red",
     xlab = "ERROR RATE",main="Histogram of ERROR RATE")
```



## Histogram of ERROR RATE



### 3.2.1 Linear regression model

In the first model, we aim to fit the data via a linear regression model. In this model, we only have one input, which is AGE, and the model attempts to find the response, which is ERROR\_RATE. This model can be formulated as:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

#### 3.2.1.1 Stan code

This is the Stan code for our linear regression model. It defines the problem in Stan syntax. In the “generated quantities” block, we draw from posterior predictive distribution to let us visualize how a manipulation of a predictor affects.

```
smodel_error_1="data {  
  int < lower = 1 > N; // Sample size  
  vector[N] x; // Predictor  
  vector[N] y; // Outcome  
}  
  
parameters {  
  real alpha; // Intercept  
  real beta; // Slope (regression coefficients)  
  real < lower = 0 > sigma; // Error SD  
}  
  
model {
```

```

y ~ normal(x * beta + alpha, sigma);
alpha ~ normal(50, 20);
beta ~ normal(1, 0.1);
}

generated quantities {
  real y_rep_error[N];
  vector[N] log_lik;
  for (n in 1:N) {
    y_rep_error[n] = normal_rng(x[n] * beta + alpha, sigma);
    log_lik[n] = normal_lpdf(y[n] | x[n] * beta + alpha, sigma);
  }
}
"
stan_error_l <- stan_model(model_code = smodel_error_l)

data_error_l <- list (x = Final$AGE,y=Final$ERROR_RATE,N = length(Final$WPM))
fit_error_l <- sampling(stan_error_l, data=data_error_l, control = list(max_treedepth = 15))

```

### 3.2.1.2 Prior distributions

We use weakly informative priors for  $\alpha \sim \text{half} - \text{normal}(0, 10)$  and  $\beta \sim \text{normal}(1, 0.1)$ . As shown in the trend plot in section ERROR\_RATE prediction, the ERROR\_RATE is non-negative. We set the sd to 10 to pick some sparse priors as well. Moreover, mean 1 and sd 0.1 were selected for the slope prior.

### 3.2.1.3 R\_hat

$\hat{R}$  is a mesearment for convergence analysis and the value below 1.1 shows stability of outcomes As<sup>^</sup>Ris below 1.1, the convergence is satisfactory. Hence, the output of our model is converged.

```
print(fit_error_l, pars=c("alpha", "beta", "sigma"), probs=c(.1,.5,.9))
```

```

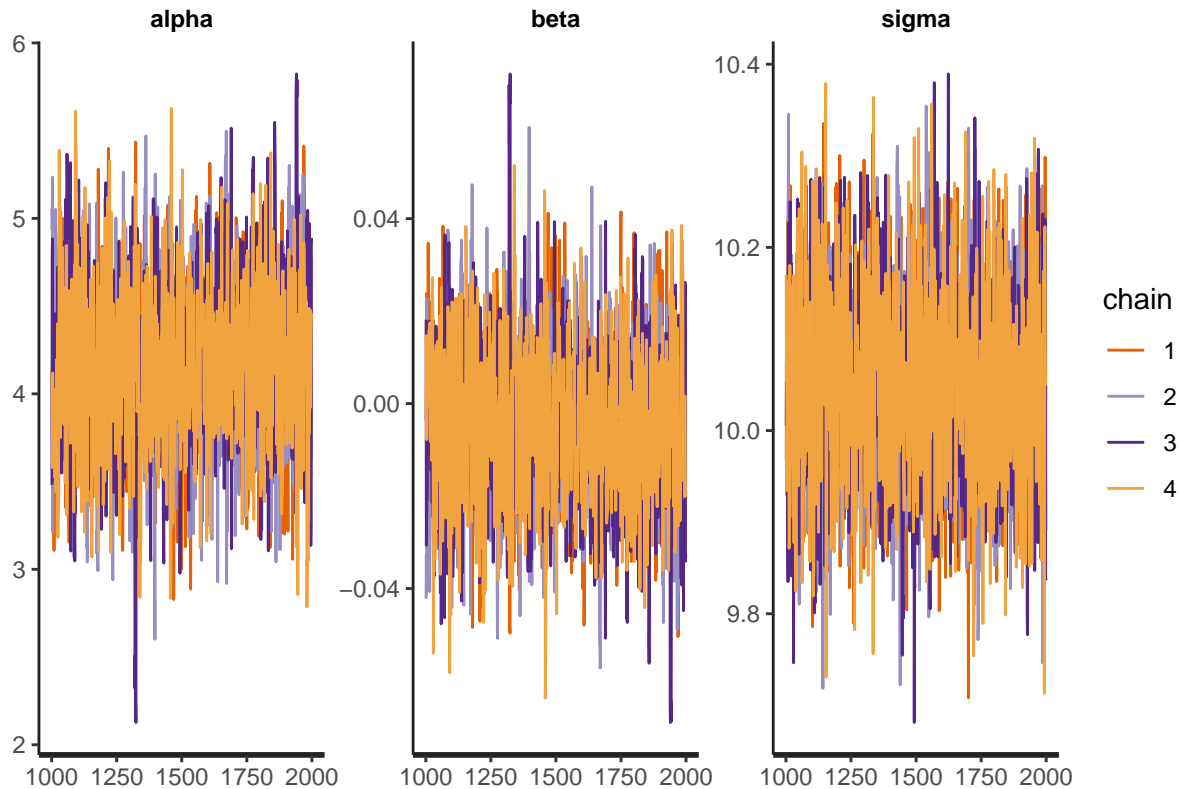
## Inference for Stan model: 0a8005e83f0024266348fe2fe04a13ec.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean   sd  10%   50%   90% n_eff Rhat
## alpha  4.16    0.01 0.46  3.59  4.16  4.74  1520    1
## beta   0.00    0.00 0.02 -0.03  0.00  0.02  1508    1
## sigma 10.05    0.00 0.10  9.92 10.05 10.18  2231    1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 20:21:44 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).

```

### 3.2.1.4 hmc convergence divergences

A useful diagnostic plot is the trace plot. It shows a time series of the Markov chains. These plots show our model is converged.

```
traceplot(fit_error_l, pars =c("alpha", "beta","sigma"))
```



### 3.2.1.5 ESS

The effective sample size (ESS) is a quantity measuring how many independent draws contain the same amount of information. Higher value of ESS is better. If the value of this parameter is small, then it shows the posterior distribution will be poor.

```
print(fit_error_l, pars=c("alpha", "beta","sigma"), probs=c(.1,.5,.9))
```

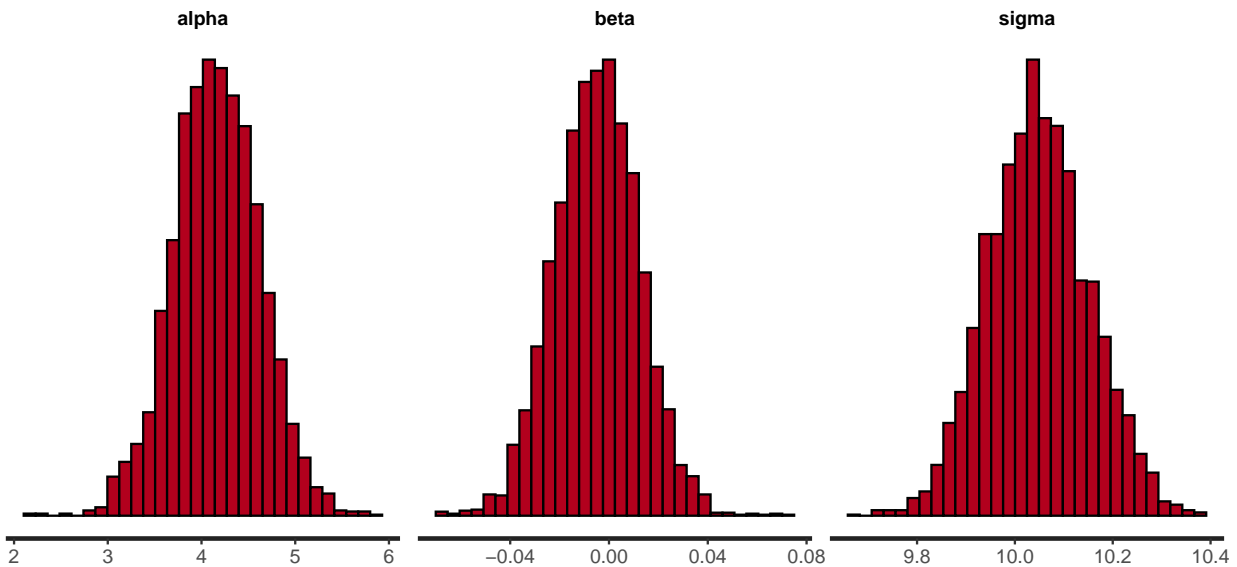
```
## Inference for Stan model: 0a8005e83f0024266348fe2fe04a13ec.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean  sd  10%  50%  90% n_eff Rhat
## alpha  4.16    0.01 0.46  3.59  4.16  4.74 1520   1
## beta   0.00    0.00 0.02 -0.03  0.00  0.02 1508   1
## sigma 10.05    0.00 0.10  9.92 10.05 10.18 2231   1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 20:21:44 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).
```

The ESS values (it shows under column `n_eff`) for our model show that it is good.

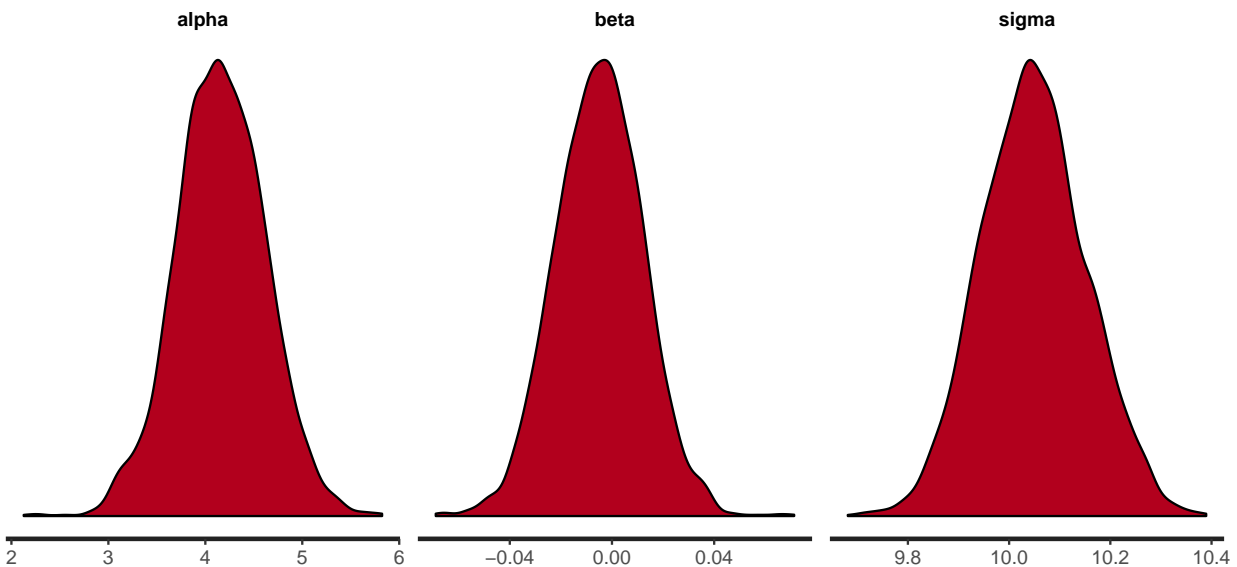
### 3.2.1.6 Posterior check

We can also look at the posterior densities & histograms

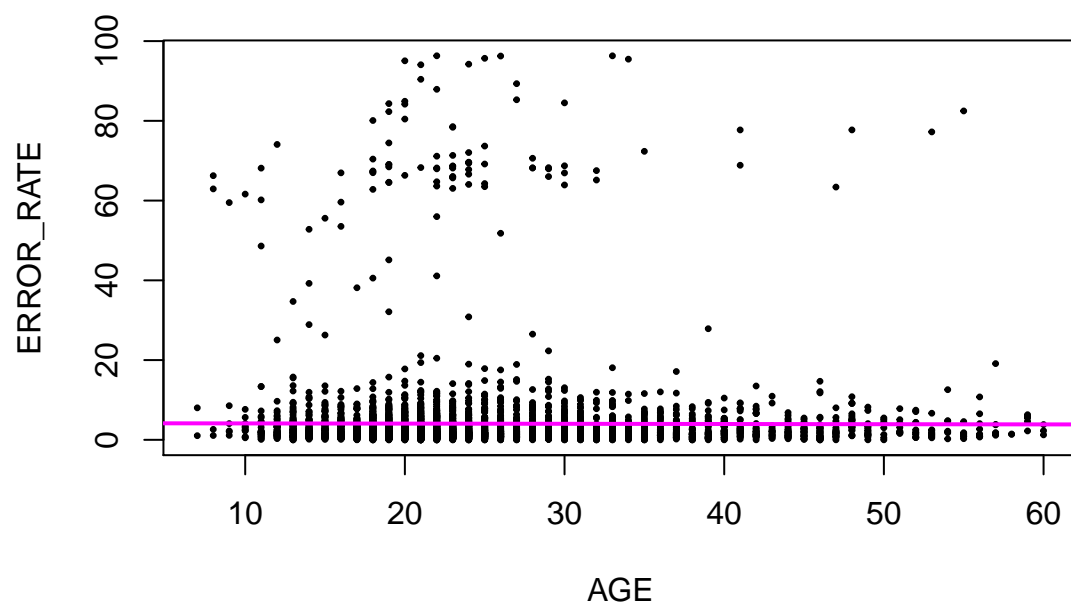
```
stan_hist(fit_error_l, pars=c("alpha", "beta", "sigma"))
```



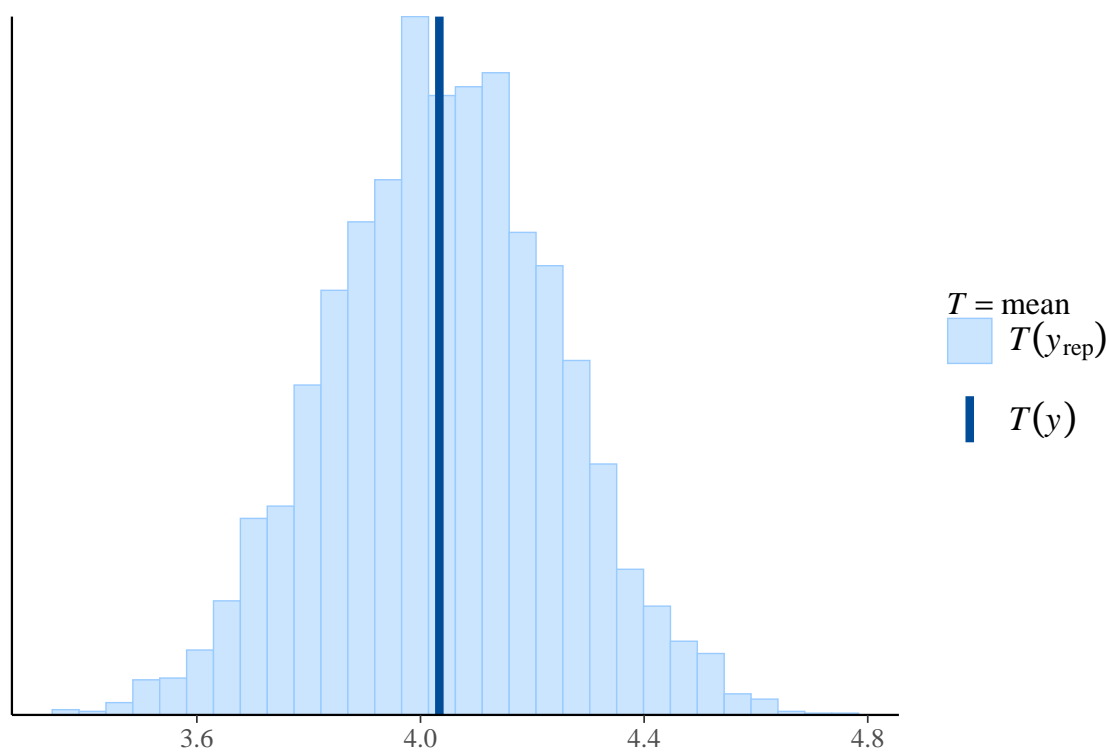
```
stan_dens(fit_error_l, pars=c("alpha", "beta", "sigma"))
```



```
posterior <- extract(fit_error_l)
plot(Final$ERROR_RATE ~ Final$AGE, pch = 20, cex = .5, xlab="AGE", ylab="ERROR_RATE", main="")
abline( mean(posterior$alpha), mean(posterior$beta), col = 6, lw = 2)
```



```
y_rep_error <- as.matrix(fit_error_l, pars = "y_rep_error")
ppc_stat(y = Final$ERROR_RATE, yrep = y_rep_error, stat = "mean")
```



These posterior plots show the model fits to the model and it is able to predict ERROR\_RATE.

### 3.2.2 Hierarchical generalized linear model

Now, we use a hierarchical generalized linear model to fit the model. This model extend linear models to include coefficients that vary by discrete groups.

$$y_i \sim \text{Normal}(\alpha_{j[i]} + x_i' \beta_{j[i]}, \sigma_{j[i]})$$

In this model, parameters are nested within one another.

#### 3.2.2.1 brms code

For this part, we used brms which provides an interface to fit our hierarchical model using Stan.

```
prior <- c(prior_string("normal(0,1)", class = "b"))
fit_error_h <- brm(ERROR_RATE ~ AGE + GENDER + (1 | USING_FEATURES),
                  data = Final, family = hurdle_gamma(), control = list(max_treedepth = 15),
                  prior = prior)
fit_error_h <- add_criterion(fit_error_h, "waic")
```

```
summary(fit_error_h)
```

```
## Warning: There were 144 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: hurdle_gamma
## Links: mu = log; shape = identity; hu = identity
## Formula: ERROR_RATE ~ AGE + GENDER + (1 | USING_FEATURES)
## Data: Final (Number of observations: 4720)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~USING_FEATURES (Number of levels: 4)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.24      0.18    0.07    0.82 1.03      110      28
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        1.45      0.13    1.20    1.72 1.02      254      501
## AGE              -0.01      0.00   -0.01   -0.00 1.01      381     1814
## GENDERmale       0.33      0.04    0.26    0.40 1.00     1058     1912
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape          0.83      0.01    0.81    0.86 1.00     2754     1775
## hu             0.01      0.00    0.01    0.02 1.01     2102     2502
##
## 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).
```

#### 3.2.2.2 Prior distributions

We use hurdle\_gamma family because there are some zero values for the response variable.

### 3.2.2.3 R\_hat

$\hat{R}$  is a mesearment for convergence analysis and the value below 1.1 shows stability of outcomes As<sup>^</sup>Ris below 1.1, the convergence is satisfactory. Hence, the output of our model is converged.

```
print(fit_error_h, pars=c("alpha", "beta", "sigma"), probs=c(.1,.5,.9))
```

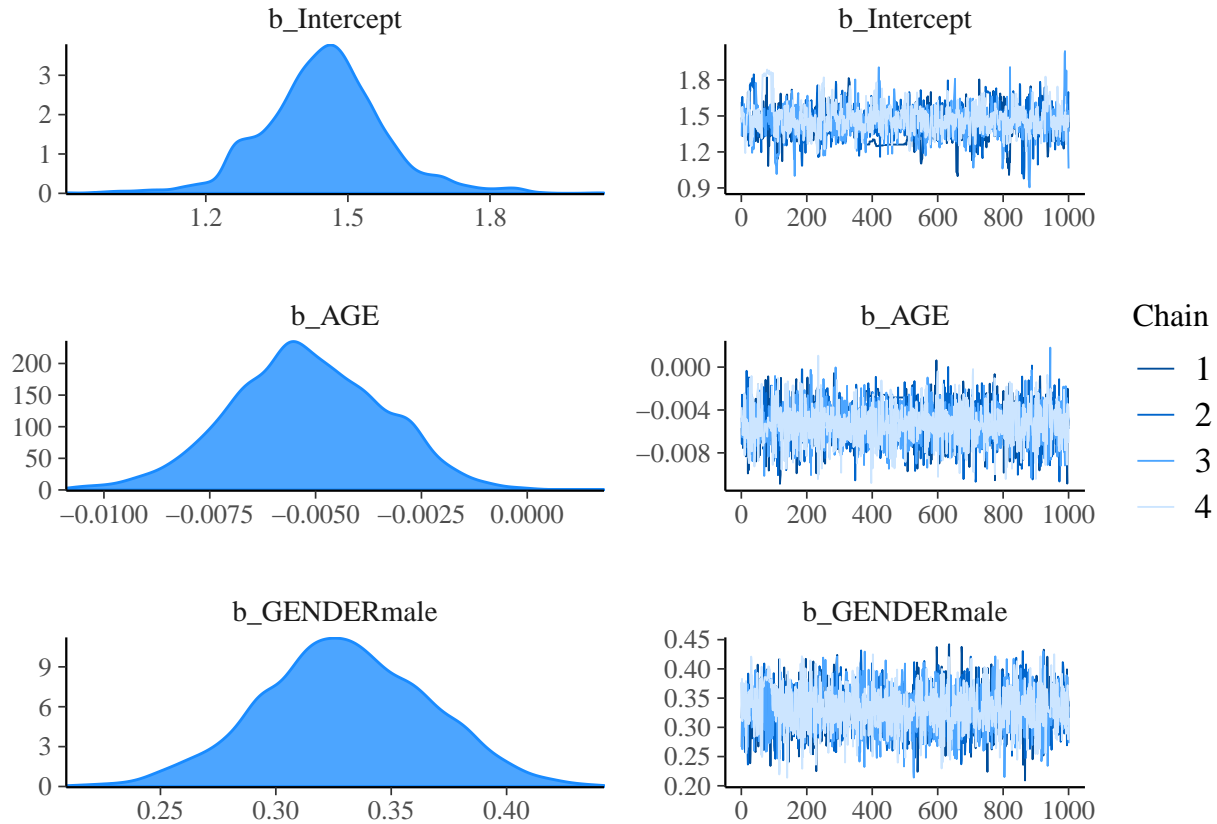
```
## Warning: There were 144 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Family: hurdle_gamma
## Links: mu = log; shape = identity; hu = identity
## Formula: ERROR_RATE ~ AGE + GENDER + (1 | USING_FEATURES)
## Data: Final (Number of observations: 4720)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~USING_FEATURES (Number of levels: 4)
##          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.24      0.18    0.07    0.82 1.03     110      28
##
## Population-Level Effects:
##          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        1.45      0.13    1.20    1.72 1.02     254     501
## AGE              -0.01      0.00   -0.01   -0.00 1.01     381    1814
## GENDERmale        0.33      0.04    0.26    0.40 1.00    1058    1912
##
## Family Specific Parameters:
##          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape          0.83      0.01    0.81    0.86 1.00     2754    1775
## hu             0.01      0.00    0.01    0.02 1.01     2102    2502
##
## 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).
```

### 3.2.2.4 hmc convergence divergences

A useful diagnostic plot is the trace plot. It shows a time series of the Markov chains. These plots show our model is converged.

```
plot(fit_error_h, pars = "~b_")
```



### 3.2.2.5 ESS

The effective sample size (ESS) is a quantity measurement showing how many independent draws contain the same amount of information. Higher value of ESS is better.

```
print(fit_error_h, pars=c("alpha", "beta", "sigma"), probs=c(.1,.5,.9))
```

```
## Warning: There were 144 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: hurdle_gamma
## Links: mu = log; shape = identity; hu = identity
## Formula: ERROR_RATE ~ AGE + GENDER + (1 | USING_FEATURES)
## Data: Final (Number of observations: 4720)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~USING_FEATURES (Number of levels: 4)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.24      0.18    0.07    0.82 1.03     110      28
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        1.45      0.13    1.20    1.72 1.02     254     501
```



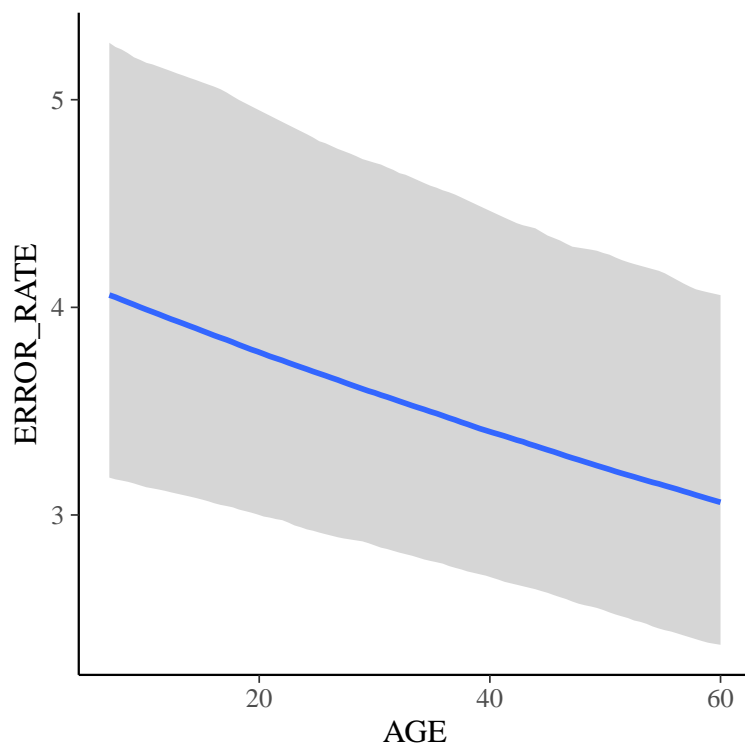
```
## AGE          -0.01      0.00     -0.01     -0.00 1.01      381      1814
## GENDERmale    0.33      0.04      0.26      0.40 1.00     1058     1912
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape      0.83      0.01      0.81      0.86 1.00      2754      1775
## hu         0.01      0.00      0.01      0.02 1.01      2102      2502
##
## 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).
```

The ESS values for our model show that it is good.

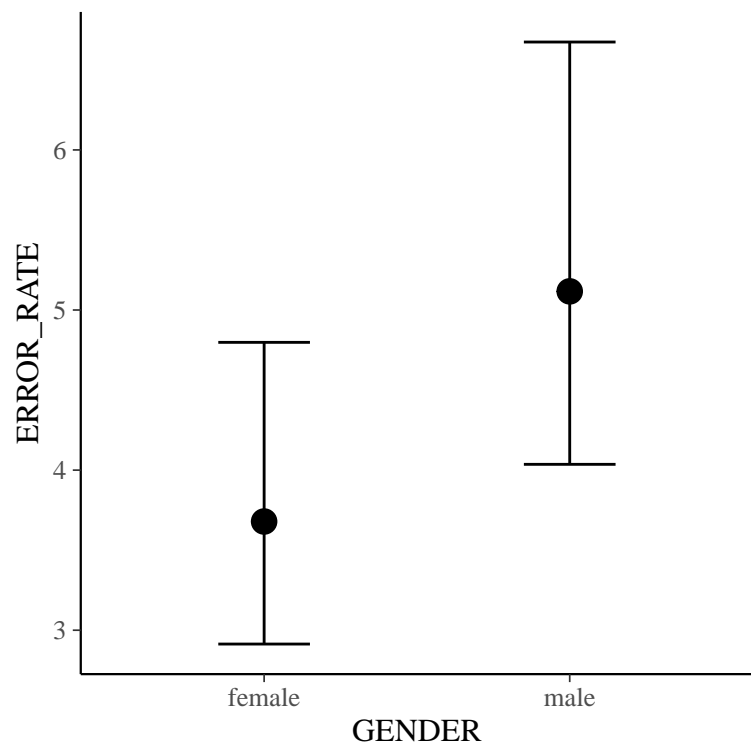
### 3.2.2.6 postrior check

We can also look at the posterior densities & histograms

```
marginal_effects(fit_error_h, effects = "AGE")
```



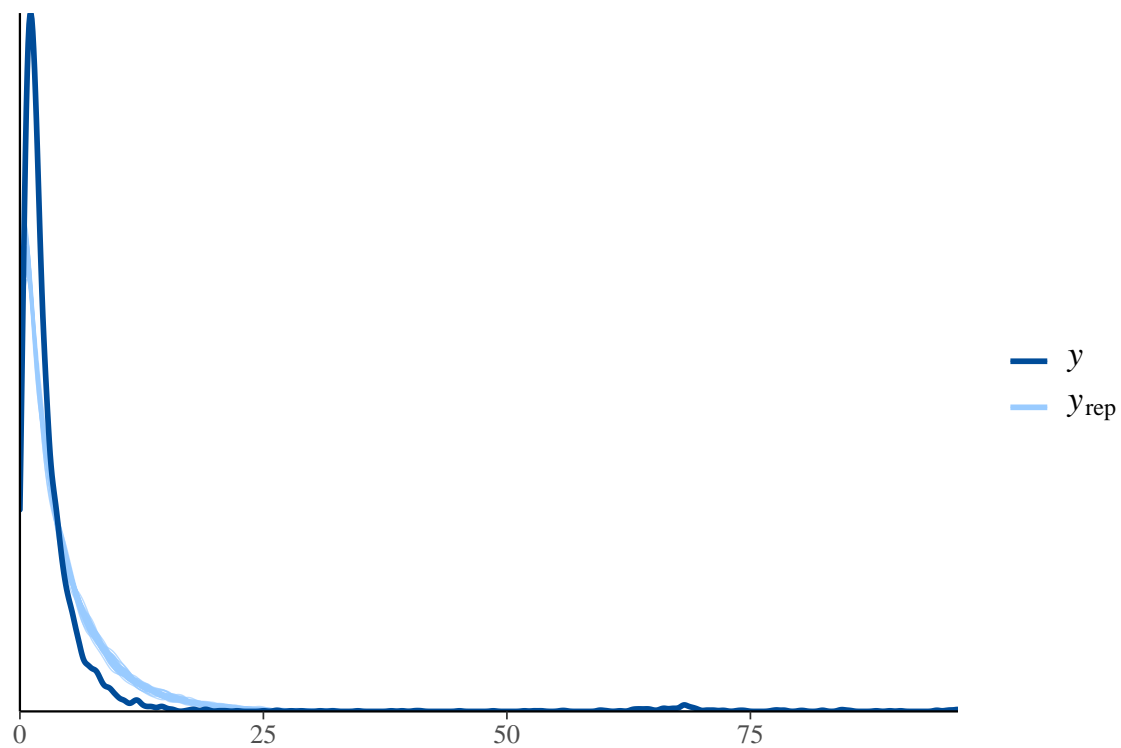
```
marginal_effects(fit_error_h, effects = "GENDER")
```



```

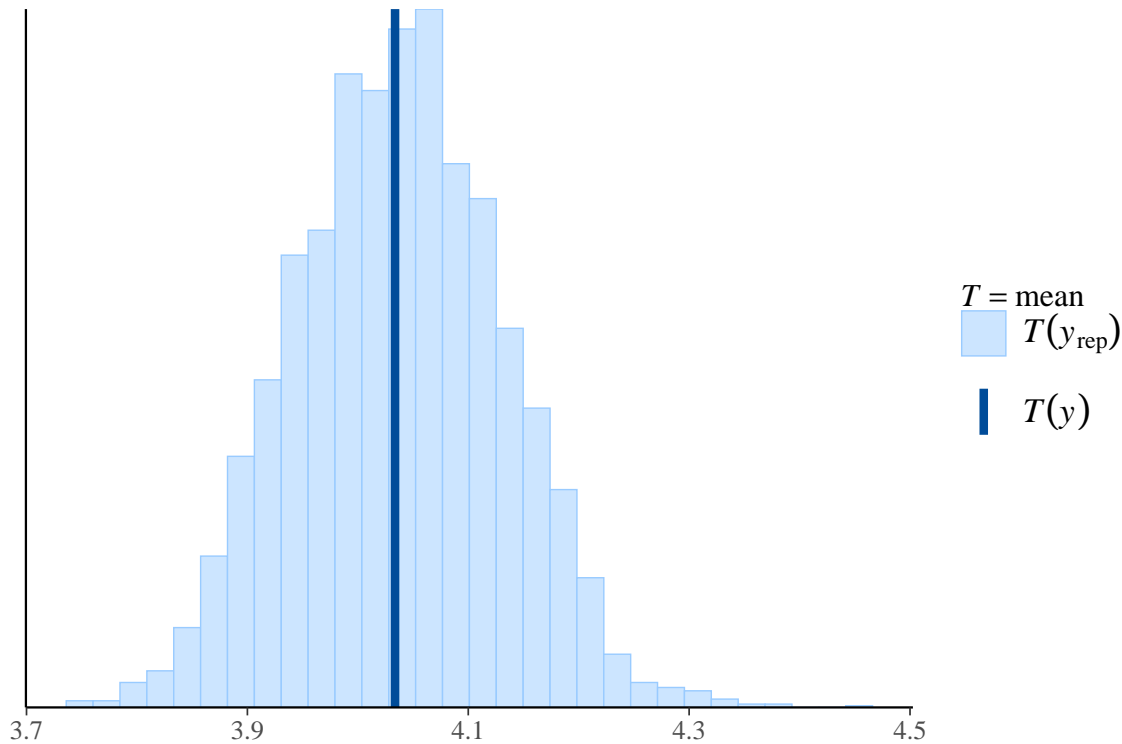
yrep_error_h <- posterior_predict(fit_error_h, draws = 500)
color_scheme_set("brightblue")
ppc_dens_overlay(Final$ERROR_RATE, yrep_error_h[1:50, ])

```



```
ppc_stat(y = Final$ERROR_RATE, yrep = yrep_error_h, stat = "mean")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



These posterior plots show the model fits to the model and it is able to predict WPM.

### 3.2.2.7 Comparison and conclusion

We used loo to compare the models. It is applied to estimate the difference in the expected predictive accuracy.

```
loo_error_1 <- loo(fit_error_1, save_psis = TRUE)
print(loo_error_1)
```

```
##
## Computed from 4000 by 4720 log-likelihood matrix
##
##      Estimate      SE
## elpd_loo -17600.7 233.1
## p_loo      24.5   3.0
## looic      35201.4 466.2
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_error_h <- loo(fit_error_h, save_psis = TRUE)
print(loo_error_h)
```

```
##
## Computed from 4000 by 4720 log-likelihood matrix
##
##           Estimate      SE
## elpd_loo -11430.7 148.0
## p_loo      33.3   3.9
## looic      22861.3 296.0
## -----
## Monte Carlo SE of elpd_loo is 0.2.
##
## Pareto k diagnostic values:
##               Count Pct.   Min. n_eff
## (-Inf, 0.5] (good)   4718 100.0%   167
## (0.5, 0.7] (ok)       2   0.0%   304
## (0.7, 1] (bad)        0   0.0%  <NA>
## (1, Inf) (very bad)  0   0.0%  <NA>
##
## All Pareto k estimates are ok (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

```
compare(loo_error_l, loo_error_h)
```

```
## elpd_diff      se
##    6170.0     115.5
```

The difference in ELPD is much larger than twice the estimated standard error again indicating that the hierarchical model is expected to have better predictive performance than the linear model.

## References

Palin, K., Feit, A.M., Kim, S., Kristensson, P.O. and Oulasvirta, A., 2019, October. How do People Type on Mobile Devices?: Observations from a Study with 37,000 Volunteers. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (p. 9). ACM.