HW 5 Solutions

Shravan Vasishth

In these exercises, you can use any or all of the four methods discussed in the lecture. In general, bridgesampling will be time-consuming; Savage-Dickey will be faster.

Exercise 1: Is there evidence for differences in the effect of cloze probability among the subjects?

Use Bayes factor to compare the centered cloze probability model, with a similar model but one that incorporates the strong assumption of no difference between subjects for the effect of centered cloze ($\tau_{u_2} = 0$).

Solution

```
data("df_eeg")
head(df_eeg)
```

```
## # A tibble: 6 x 6
##    subj cloze  item  n400 cloze_ans     N
##   <dbl> <dbl> <dbl> <dbl>     <dbl> <dbl>
## 1     1  0        1  7.08         0    44
## 2     1  0.03     2 -0.68         1    44
## 3     1  1        3  1.39        44    44
## # i 3 more rows
```

```
data(df_eeg)
df_eeg <- df_eeg %>%
  mutate(c_cloze = cloze - mean(cloze))
```

```
priorsFULL <-
  c(prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b, coef = c_cloze),
    prior(normal(0, 50), class = sigma),
    prior(normal(0, 20), class = sd, coef = Intercept, group = subj),
    prior(normal(0, 5), class = sd, coef = c_cloze, group = subj),
    prior(normal(0, 20), class = sd, coef = Intercept, group = item),
    prior(normal(0, 5), class = sd, coef = c_cloze, group = item),
     prior(lkj(2), class = cor))

priorsNULL <-
  c(prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b, coef = c_cloze),
    prior(normal(0, 50), class = sigma),
    prior(normal(0, 20), class = sd, coef = Intercept, group = subj),
    #prior(normal(0, 5), class = sd, coef = c_cloze, group = subj),
    prior(normal(0, 20), class = sd, coef = Intercept, group = item),
```

```r
    prior(normal(0, 5), class = sd, coef = c_cloze, group = item),
     prior(lkj(2), class = cor, group = item))
```

Using bridge sampling:

```r
fit_N400_h <- brm(n400 ~ c_cloze +
                    (c_cloze | subj) + (c_cloze | item),
                  prior = priorsFULL,
                  warmup = 2000,
                  iter = 20000,
                  control = list(adapt_delta = 0.9),
                  save_pars = save_pars(all = TRUE),
                  data = df_eeg)

fit_N400_h_NULL <- brm(n400 ~ c_cloze +
                    (1 | subj) + (c_cloze | item),
                  prior = priorsNULL,
                  warmup = 2000,
                  iter = 20000,
                  control = list(adapt_delta = 0.9),
                  save_pars = save_pars(all = TRUE),
                  data = df_eeg)
```

```r
bayes_factor(fit_N400_h,fit_N400_h_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of fit_N400_h over fit_N400_h_NULL: 0.49845
```

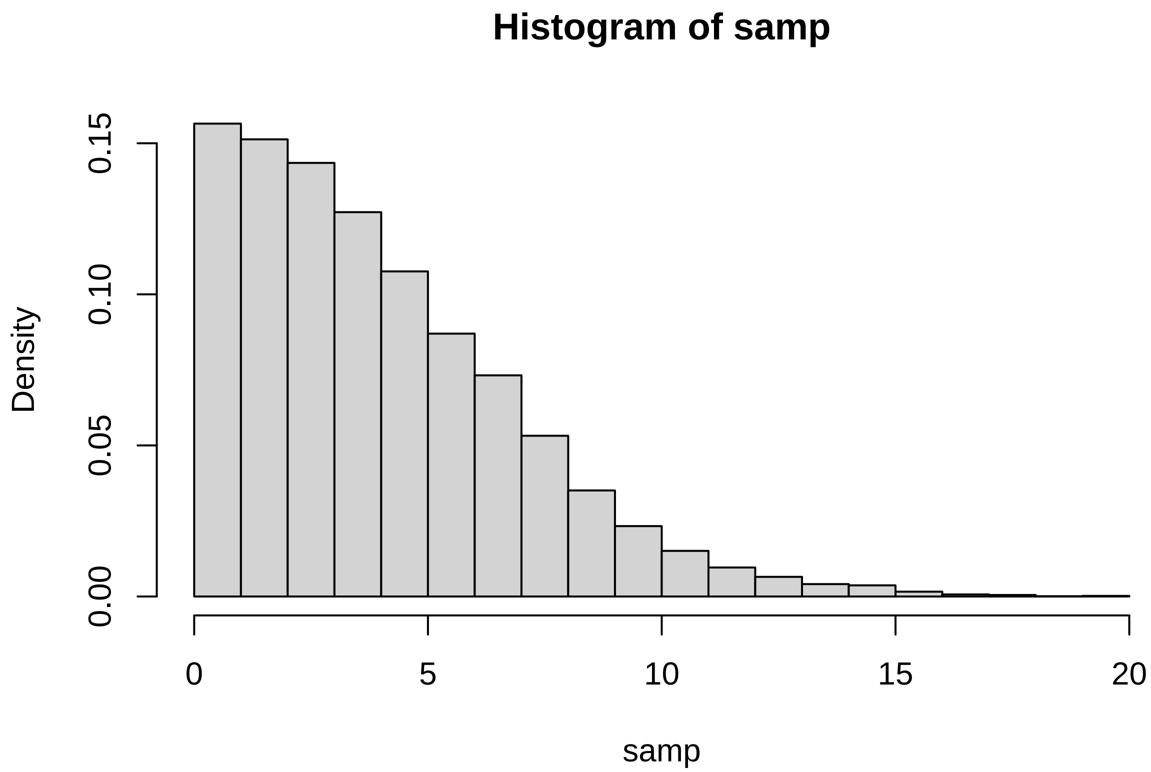
```r
## check for stability:
bayes_factor(fit_N400_h,fit_N400_h_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
```

```
## Estimated Bayes factor in favor of fit_N400_h over fit_N400_h_NULL: 0.50163
```

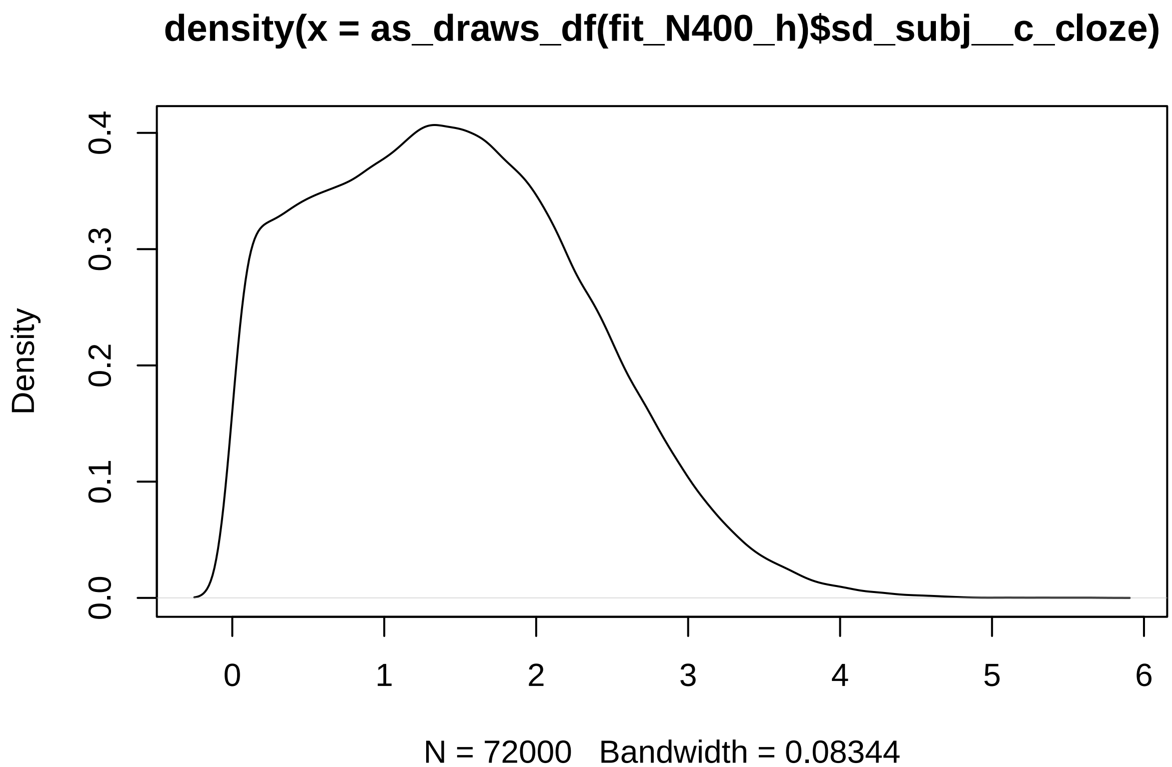
Using Savage-Dickey

```r
## Simulate prior:
samp<-extraDistr::rtnorm(10000,mean=0,sd=5,a=0)
hist(samp,freq=FALSE)
```




```r
## twice that of the untruncated normal because
## the truncated normal has to be renormalized:
prior_dens_0<-dnorm(0,mean=0,sd=5)*2

plot(density(as_draws_df(fit_N400_h)$sd_subj__c_cloze))
```




```r
post_dens_0<-0.325 ## eyeballed from histogram

## evidence for full model:
prior_dens_0/post_dens_0
```

```
## [1] 0.491
```

Conclusion: No evidence for between-subject variability in c_cloze. But the result from the Bayes factor is **inconclusive**. We can't say that there is evidence against between-subject variability in c_cloze. Be careful about wording and conclusions.

Compare with lmer:

```r
library(lme4)
mFULL<-lmer(n400 ~ c_cloze +
                        (1 + c_cloze || subj) + (c_cloze | item),
                      data = df_eeg)
mNULL<-lmer(n400 ~ c_cloze +
                        (1 | subj) + (c_cloze | item),
                      data = df_eeg)

anova(mFULL,mNULL)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_eeg
## Models:
## mNULL: n400 ~ c_cloze + (1 | subj) + (c_cloze | item)
## mFULL: n400 ~ c_cloze + ((1 | subj) + (0 + c_cloze | subj)) + (c_cloze | item)
##       npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## mNULL    7 22216 22258 -11101    22202
## mFULL    8 22217 22265 -11101    22201  0.99  1       0.32
```

It would be a mistake to conclude that there is evidence against between subject variability in c_cloze.

Exercise 2: Is there evidence for the claim that English subject relative clauses are easier to process than object relative clauses?

Consider again the reading time data coming from Experiment 1 of @grodner presented in exercise @ref(exr:hierarchical-logn):

```
data("df_gg05_rc")
df_gg05_rc
```

```
## # A tibble: 672 x 7
##    subj  item condition    RT residRT qcorrect experiment
##   <int> <int> <chr>     <int>   <dbl>    <int> <chr>
## 1     1     1 objgap      320   -21.4        0 tedrg3
## 2     1     2 subjgap     424    74.7        1 tedrg2
## 3     1     3 objgap      309   -40.3        0 tedrg3
## # i 669 more rows
```

You should use a sum coding for the predictors. Here, object relative clauses (`"objgaps"`) are coded $+1/2$, and subject relative clauses as $-1/2$.

```
df_gg05_rc <- df_gg05_rc %>%
  mutate(c_cond = if_else(condition == "objgap", 1/2, -1/2))
head(df_gg05_rc)
```

```
## # A tibble: 6 x 8
##    subj  item condition    RT residRT qcorrect experiment c_cond
##   <int> <int> <chr>     <int>   <dbl>    <int> <chr>       <dbl>
## 1     1     1 objgap      320   -21.4        0 tedrg3        0.5
## 2     1     2 subjgap     424    74.7        1 tedrg2       -0.5
## 3     1     3 objgap      309   -40.3        0 tedrg3        0.5
## # i 3 more rows
```

Using the bayes_factor function discussed in class, quantify the evidence against the null model (no population-level reading time difference between SRC and ORC) relative to the following alternative models:

a. $\beta \sim Normal(0, 1)$
b. $\beta \sim Normal(0, 0.1)$
c. $\beta \sim Normal(0, 0.01)$
d. $\beta \sim Normal_+(0, 1)$
e. $\beta \sim Normal_+(0, 0.1)$
f. $\beta \sim Normal_+(0, 0.01)$

(A $Normal_+(.)$ prior can be set in `brms` by defining a lower boundary as 0, with the argument `lb = 0`.)

What are the Bayes factors in favor of the alternative models a-f, compared to the null model?

Now carry out a standard frequentist likelihood ratio test using the `anova()` function that is used with the `lmer()` function. The commands for doing this comparison would be:

```
m_full <- lmer(log(RT) ~ c_cond +
                 (c_cond || subj) + (c_cond || item),
               df_gg05_rc)
m_null <- lmer(log(RT) ~ 1 + (c_cond||subj) + (c_cond || item),
               df_gg05_rc)
anova(m_null, m_full)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_gg05_rc
## Models:
## m_null: log(RT) ~ 1 + ((1 | subj) + (0 + c_cond | subj)) + ((1 | item) + (0 + c_cond | item))
## m_full: log(RT) ~ c_cond + ((1 | subj) + (0 + c_cond | subj)) + ((1 | item) + (0 + c_cond | item))
##        npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)
## m_null    6 714 741   -351      702
## m_full    7 710 741   -348      696  6.12  1      0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How do the conclusions from the Bayes factor analyses compare with the conclusion we obtain from the frequentist model comparison?

Solution

First, let's fit all the models with the different priors. Each time we compute a Bayes factor, we save the result after making sure that the BF calculation is stable.

Normal(0,1)

```
priorNULL <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))

priorN01 <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0,1), class = b, coef=c_cond),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))
```

```
fit_gg05_n01 <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                     prior = priorN01,
                     family=lognormal(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)

fit_gg05_NULL <- brm(RT ~ 1 + (c_cond | subj) + (c_cond | item),
                     prior = priorNULL,
                     family=lognormal(),
                     save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_n01,fit_gg05_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of fit_gg05_n01 over fit_gg05_NULL: 0.89696
```

```r
(bf_n01<-bayes_factor(fit_gg05_n01,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
```

```
## [1] 0.903
```

Normal(0.0.1)

```r
priorn0_1 <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0,0.1), class = b, coef=c_cond),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))
```

```r
fit_gg05_n0_1 <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                     prior = priorn0_1,
                     family=lognormal(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)
```

```r
bayes_factor(fit_gg05_n0_1,fit_gg05_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of fit_gg05_n0_1 over fit_gg05_NULL: 4.41698
```

```r
(bf_n0_1<-bayes_factor(fit_gg05_n0_1,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## [1] 4.38
```

Normal(0.01)

```
priorn0_01 <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0,0.01), class = b, coef=c_cond),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))
```

```
fit_gg05_n0_01 <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                     prior = priorn0_01,
                     family=lognormal(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_n0_01,fit_gg05_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
```

```
## Estimated Bayes factor in favor of fit_gg05_n0_01 over fit_gg05_NULL: 1.08085
```

```
(bf_n0_01<-bayes_factor(fit_gg05_n0_01,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
```

```
## [1] 1.08
```

Normal+(0,1)

```
priorN01tr <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0,1), class = b, lb = 0),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))
```

```
fit_gg05_n01tr <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                     prior = priorN01tr,
                     family=lognormal(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_n01tr,fit_gg05_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of fit_gg05_n01tr over fit_gg05_NULL: 1.78004
```

```
(bf_n01tr<-bayes_factor(fit_gg05_n01tr,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## [1] 1.79
```

Normal+(0.0.1)

```
priorn0_1tr <- c(prior(normal(0,10), class = Intercept),
                prior(normal(0,0.1), class = b, lb=0),
                prior(normal(0, 1), class = sd),
                prior(normal(0, 1), class = sigma),
                prior(lkj(2), class = cor))
```

```
fit_gg05_n0_1tr <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                     prior = priorn0_1tr,
                     family=lognormal(),
```

```
                     save_pars = save_pars(all = TRUE),
                      warmup=2000,
                      iter=20000,
                      data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_n0_1tr,fit_gg05_NULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5

## Estimated Bayes factor in favor of fit_gg05_n0_1tr over fit_gg05_NULL: 8.72076
```

```
(bf_n0_1tr<-bayes_factor(fit_gg05_n0_1tr,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5

## [1] 8.74
```

Normal+(0.01)

```
priorn0_01tr <- c(prior(normal(0,10), class = Intercept),
                 prior(normal(0,0.01), class = b,lb=0),
                 prior(normal(0, 1), class = sd),
                 prior(normal(0, 1), class = sigma),
                 prior(lkj(2), class = cor))
```

```
fit_gg05_n0_01tr <- brm(RT ~ c_cond + (c_cond | subj) + (c_cond | item),
                      prior = priorn0_01tr,
                      family=lognormal(),
                     save_pars = save_pars(all = TRUE),
                      warmup=2000,
                      iter=20000,
                      data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_n0_01tr,fit_gg05_NULL)
```

```
## Iteration: 1
```

```
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of fit_gg05_n0_01tr over fit_gg05_NULL: 1.41143
```

```r
(bf_n0_01tr<-bayes_factor(fit_gg05_n0_01tr,fit_gg05_NULL)$bf)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## [1] 1.44
```

Summary

```r
bfs<-c(bf_n01,bf_n0_1,bf_n0_01,bf_n01tr,bf_n0_1tr,bf_n0_01tr)

pr<-c("N(0,1)","N(0.0.1)","N(0.0.01)","N+(0,1)","N+(0.0.1)","N+(0.0.01)")

data.frame(priors=pr,bf=bfs)
```

```
##       priors    bf
## 1     N(0,1) 0.903
## 2   N(0.0.1) 4.381
## 3  N(0.0.01) 1.078
## 4    N+(0,1) 1.791
## 5  N+(0.0.1) 8.744
## 6 N+(0.0.01) 1.437
```

What we see is that under the assumption that the prior on the target parameter is Normal(0,0.1), we have some evidence for the effect. The way I would interpret this is that under the assumption that the prior plausible range of values lie between ± 85 ms with 95% probability:

```r
mean(df_gg05_rc$RT)
```

```
## [1] 420
```

```r
exp(6.05 + 0.2/2) - exp(6.05 - 0.2/2)
```

```
## [1] 85
```

```r
exp(6.05 + -0.2/2) - exp(6.05 - -0.2/2)
```

```
## [1] -85
```

we can conclude that we have some evidence for the RC effect.

I just asked for the BFs with truncated priors, but usually we would not make such a strong assumption that we are 100% sure that the effect is positive (we can never really be sure). But in theory there could be situations where a truncated prior is justifiable, so I included an example. of that here.

By contrast, the frequentist analysis gives a somewhat misleading conclusion: strong evidence for the effect. For more discussion, see:

- Shravan Vasishth and Andrew Gelman. How to embrace variation and accept uncertainty in linguistic and psycholinguistic data analysis. Linguistics, 59:1311–1342, 2021
- Shravan Vasishth, Himanshu Yadav, Daniel Schad, and Bruno Nicenboim. Sample size determination for Bayesian hierarchical models commonly used in psycholinguistics. Computational Brain and Behavior, 2022.
- Shravan Vasishth. Some right ways to analyze (psycho)linguistic data. Annual Review of Linguistics, 9:273–291, 2023.

Exercise 3: In the Grodner and Gibson 2005 data, in question-response accuracies, is there evidence for the claim that sentences with subject relative clauses are easier to comprehend?

Assume here that for the effect of RC on question accuracy, $\beta \sim Normal(0, 0.1)$ is a reasonable prior, and that for all the variance components, the same prior, $\tau \sim Normal_+(0, 1)$, is a reasonable prior.

Consider the question response accuracy of the data of Experiment 1 of Grodner and Gibson 2005.

a. Compare a model that assumes that RC type affects question accuracy on the population level and with the effect varying by-subjects and by-items with *a null model* that assumes that there is no population-level effect present.

b. Compare a model that assumes that RC type affects question accuracy on the population level, where the effect varies by-subjects and by-items with *another null model* that assumes that there is no population-level or group-level effect present, that is, that there is no overall effect and there are no by-subject or by-item effects. What's the meaning of the results of the Bayes factor analysis?

Solution

Descriptively, the mean accuracy for the object gap condition is a bit higher (surprising! It should have been lower).

```
head(df_gg05_rc)
```

```
## # A tibble: 6 x 8
##    subj  item condition    RT residRT qcorrect experiment c_cond
##   <int> <int> <chr>     <int>   <dbl>    <int> <chr>       <dbl>
## 1     1     1 objgap      320   -21.4        0 tedrg3        0.5
## 2     1     2 subjgap     424    74.7        1 tedrg2       -0.5
## 3     1     3 objgap      309   -40.3        0 tedrg3        0.5
## # i 3 more rows
```

```
with(df_gg05_rc,tapply(qcorrect,condition,mean))
```

```
##  objgap subjgap
##   0.884   0.848
```

(a)

```r
priors<-c(prior(normal(0,10), class = Intercept),
                prior(normal(0,0.1), class = b, coef=c_cond),
                prior(normal(0, 1), class = sd),
                prior(lkj(2), class = cor))

priorsNULL <- c(prior(normal(0,10), class = Intercept),
                #prior(normal(0,0.1), class = b, coef=c_cond),
                prior(normal(0, 1), class = sd),
                prior(lkj(2), class = cor))
```

```r
fit_gg05_acc <- brm(qcorrect ~ c_cond + (c_cond | subj) +
                      (c_cond | item),
                     prior = priors,
                     family=bernoulli(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)

fit_gg05_accNULL <- brm(qcorrect ~ c_cond + (c_cond | subj) +
                      (c_cond | item),
                     prior = priorsNULL,
                     family=bernoulli(),
                    save_pars = save_pars(all = TRUE),
                     warmup=2000,
                     iter=20000,
                     data = df_gg05_rc)
```

```r
bayes_factor(fit_gg05_acc,fit_gg05_accNULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

## Estimated Bayes factor in favor of fit_gg05_acc over fit_gg05_accNULL: 0.78078
```

```r
bayes_factor(fit_gg05_acc,fit_gg05_accNULL)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 1
```

```
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

## Estimated Bayes factor in favor of fit_gg05_acc over fit_gg05_accNULL: 0.77469
```

Inconclusive.

(b)

```
priorsNULL2 <- c(prior(normal(0,10), class = Intercept)
                 #prior(normal(0,0.1), class = b, coef=c_cond),
                 # prior(normal(0, 1), class = sd),
                 #prior(lkj(2), class = cor)
                 )
```

```
fit_gg05_accNULL2 <- brm(qcorrect ~ 1,
                      prior = priorsNULL2,
                      family=bernoulli(),
                     save_pars = save_pars(all = TRUE),
                      warmup=2000,
                      iter=20000,
                      data = df_gg05_rc)
```

```
bayes_factor(fit_gg05_acc,fit_gg05_accNULL2)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4

## Estimated Bayes factor in favor of fit_gg05_acc over fit_gg05_accNULL2: 1583527207.07116
```

```
bayes_factor(fit_gg05_acc,fit_gg05_accNULL2)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4

## Estimated Bayes factor in favor of fit_gg05_acc over fit_gg05_accNULL2: 1580281897.82356
```

The massive BF is arising because our null model is overly simple; it's easy for the full model to achieve a much higher marginal likelihood. This isn't a comparison one would ever do in practice.