How to embrace variation and accept uncertainty in linguistic and psycholinguistic data

Shravan Vasishth and Andrew Gelman

3/19/2022

I assume that the reader has at least glanced at the paper (https://www.degruyter.com/document/doi/10.1515/ling-2019-0051/html) before working through this file. Otherwise the plots will make no sense.

If there are any problems in running this code, please contact Shravan Vasishth.

Set up libraries

```
## ----setup,include=FALSE,cache=FALSE,echo=FALSE-------------------------
library(MASS)
library(knitr)
library(xtable)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
#library(sjPlot)
library(rstan)
```

```
## Loading required package: StanHeaders

## rstan (Version 2.21.3, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

##
## Attaching package: 'rstan'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
options(mc.cores = parallel::detectCores())
library(brms)
```

```
## Loading required package: Rcpp

## Loading 'brms' package (version 2.16.3). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').

##
## Attaching package: 'brms'

## The following object is masked from 'package:rstan':
##
##     loo

## The following object is masked from 'package:stats':
##
##     ar
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.8.1

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(ggridges)
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'lme4'

## The following object is masked from 'package:brms':
##
##     ngrps
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(magick)
```

```
## Linking to ImageMagick 6.9.12.3
## Enabled features: cairo, fontconfig, freetype, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fftw, ghostscript, x11
```

```
library(xtable)

#theme_set(theme_apa())

# set global chunk options, put figures into folder
options(warn=-1, replace.assign=TRUE)
opts_chunk$set(fig.path='figures/figure-', fig.align='center', fig.show='hold')
options(replace.assign=TRUE,width=75)
opts_chunk$set(dev='postscript')
opts_chunk$set(echo = TRUE,purl=TRUE)

## ----loadfunctions, include=TRUE, echo=FALSE, warning=FALSE, error=TRUE, message=TRUE----
source("../R/createStanDat.R")
source("../R/createStanDatAcc.R")
source("../R/magnifytext.R")
source("../R/multiplot.R")
source("../R/plotresults.R")
source("../R/plotpredictions.R")
source("../R/stan_results.R")
```

Load existing data on agreement attraction.

```
## ----agrmtattrn,echo=FALSE,warning=FALSE,message=FALSE-------------------
#Dillon et al 2013 Expt 1
DillonE1<-read.table("../data/DillonE1.txt",header=TRUE)
#Lago et al 2015 data (all expts):
```

```
Lago<-read.csv("../data/Lago.csv",header=T)
##Wagers et al 2009 data (all expts):
load("../data/Wagers.Rdata")
load("../data/Tucker.RData")

#Dillon E1:
DillonE1$cond<-factor(DillonE1$cond)
DillonE1Mism<-subset(DillonE1,fixationtype=="tt" & region==5 & cond%in%c(3,4) & value!="NA")
DillonE1Mism$cond<-factor(DillonE1Mism$cond)
DillonE1Mism$int<-ifelse(DillonE1Mism$cond==3,"low","high")
DillonE1Mism$x<-ifelse(DillonE1Mism$cond==3,-1,1)
dillonE1<-DillonE1Mism[,c(1,3,4,14,15)]
dillonE1$expt<-factor("dillonE1")
colnames(dillonE1)[3]<-"rt"

nsubj_dillonE1<-length(unique(dillonE1$subj))

#get critical region data:
#dillonE <- subset(dillonE1, roi==5 & cond!='filler')

##Lago:
dat<-Lago
#critical region: not used because published paper found
#significant effects in postcrit region only
e1<-subset(dat,Experiment=="Experiment1" & Region=="06v1")
e2<-subset(dat,Experiment=="Experiment2" & Region=="06aux")
e3a<-subset(dat,Experiment=="Experiment3A" & Region=="06aux")
e3b<-subset(dat,Experiment=="Experiment3B" & Region=="aux")

nsubj_lagoe1<-length(unique(e1$Subject))
nsubj_lagoe2<-length(unique(e2$Subject))
nsubj_lagoe3a<-length(unique(e3a$Subject))
nsubj_lagoe3b<-length(unique(e3b$Subject))


#postcritical region:
poste1<-subset(dat,Experiment=="Experiment1" & Region=="07prep")
poste2<-subset(dat,Experiment=="Experiment2" & Region=="07adv")
poste3a<-subset(dat,Experiment=="Experiment3A" & Region=="07a")
poste3b<-subset(dat,Experiment=="Experiment3B" & Region=="a")

##e1: a,b
#-(a) Ungram , singular attractor (interference condition)
#La *nota* que la chica escribieron en la clase alegrC3 a su amiga
#The note that the girl wrotepl during class cheered her friend up
#-(b) Ungram , plural attractor (baseline condition)
#Las *notas* que la chica escribieron en la clase alegraron a su amiga
#The notes that the girl wrotepl during class cheered her friend up
poste1<-subset(poste1,Condition%in%c("a","b"))
poste1$Condition<-factor(poste1$Condition)
poste1$x<-ifelse(poste1$Condition=="a",-1,1)
poste1$int<-ifelse(poste1$Condition=="a","low","high")
poste1<-poste1[,c(1,3,8,15,14)]
```

```
poste1$expt<-factor("lagoE1")
lagoE1<-poste1
colnames(lagoE1)<-c("subj","item","rt","int","x","expt")

#e2: c,d
poste2<-subset(poste2,Condition%in%c("c","d"))
poste2$Condition<-factor(poste2$Condition)
poste2$x<-ifelse(poste2$Condition=="c",-1,1)
poste2$int<-ifelse(poste2$Condition=="c","low","high")
#head(poste2)
poste2<-poste2[,c(1,3,8,15,14)]
poste2$expt<-factor("lagoE2")
lagoE2<-poste2
colnames(lagoE2)<-c("subj","item","rt","int","x","expt")

#e3a: e,f
poste3a<-subset(poste3a,Condition%in%c("e","f"))
poste3a$Condition<-factor(poste3a$Condition)
#-(e) Ungram, singular attractor (interference condition)
#La *nota* que la chica van a escribir en la clase alegrarC! a su amiga
#The note that the girl are going to write during class will cheer her friend up
#-(f) Ungram, plural attractor (baseline condition)
#Las *notas* que la chica van a escribir en la clase alegrarC!n a su amiga
#The notes that the girl are going to write during class will cheer her friend up
#boxplot(RT~Condition,poste3a)
poste3a$x<-ifelse(poste3a$Condition=="e",-1,1)
poste3a$int<-ifelse(poste3a$Condition=="e","low","high")
poste3a<-poste3a[,c(1,3,8,15,14)]
poste3a$expt<-factor("lagoE3a")
lagoE3a<-poste3a
colnames(lagoE3a)<-c("subj","item","rt","int","x","expt")

#e3b: e,f
poste3b<-subset(poste3b,Condition%in%c("e","f"))
poste3b$Condition<-factor(poste3b$Condition)
#-(e) Ungram, singular attractor (baseline condition)
#The player that the coach were always praising very enthusiastically decided to     leave the team
#-(f) Ungram, plural attractor (interference condition)
#The players that the coach were always praising very enthusiastically decided to     leave the team
poste3b$x<-ifelse(poste3b$Condition=="e",-1,1)
poste3b$int<-ifelse(poste3b$Condition=="e","low","high")
poste3b<-poste3b[,c(1,3,8,15,14)]
poste3b$expt<-factor("lagoE3b")
lagoE3b<-poste3b
colnames(lagoE3b)<-c("subj","item","rt","int","x","expt")

#Wagers:
E2postcrit<-subset(Experiment2,Region==7)
nsubj_wagerse2<-length(unique(E2postcrit$Subj))
#E2$intr.au<-ifelse(E2$rchead=="pl" & E2$gramm=="ungram",1/2,
#                   ifelse(E2$rchead=="sg" & E2$gramm=="ungram",-1/2,
#                          0))
#d (sing),h (plu)
```

```
#unique(subset(E2postcrit,gramm=="ungram")$Condition)
E2postcrit<-subset(E2postcrit,Condition%in%c("d","h"))
E2postcrit$Condition<-factor(E2postcrit$Condition)
E2postcrit$x<-ifelse(E2postcrit$Condition=="d",-1,1)
E2postcrit$int<-ifelse(E2postcrit$Condition=="d","low","high")
#colnames(E2postcrit)
E2postcrit<-E2postcrit[,c(4,3,8,13,12)]
E2postcrit$expt<-factor("wagersE2")
wagersE2<-E2postcrit
colnames(wagersE2)<-c("subj","item","rt","int","x","expt")

#E3
E3postcrit<-subset(Experiment3,Region==7)
nsubj_wagerse3<-length(unique(E3postcrit$Subj))
#E3crit$intr.au.pl<-ifelse(E3crit$gramm=="ungram" & E3crit$rcsubj=="sg" &
#                            E3crit$rchead=="pl",1/2,
#                         ifelse(E3crit$gramm=="ungram" & E3crit$rcsubj=="sg" &
#                                   E3crit$rchead=="sg",-1/2,0))

#E3crit$intr.au.sg<-ifelse(E3crit$gramm=="ungram" & E3crit$rcsubj=="pl" &
#                            E3crit$rchead=="sg",1/2,
#                          ifelse(E3crit$gramm=="ungram" & E3crit$rcsubj=="pl" &
#                                   E3crit$rchead=="pl",-1/2,0))

E3postcrit_pl<-subset(E3postcrit,gramm=="ungram" & rcsubj=="sg")
E3postcrit_pl$Condition<-factor(E3postcrit_pl$Condition)
E3postcrit_sg<-subset(E3postcrit,gramm=="ungram" & rcsubj=="pl")
E3postcrit_sg$Condition<-factor(E3postcrit_sg$Condition)

#unique(E3postcrit_pl$Condition) b,f
#unique(E3postcrit_sg$Condition) c,g

#head(subset(E3postcrit_sg,rchead=="sg"))
#head(E3postcrit_pl)

##plural:
E3postcrit_pl$x<-ifelse(E3postcrit_pl$Condition=="b",-1,1)
E3postcrit_pl$int<-ifelse(E3postcrit_pl$Condition=="b","low","high")
E3postcrit_pl<-E3postcrit_pl[,c(4,3,8,15,14)]
E3postcrit_pl$expt<-factor("wagersE3pl")
colnames(E3postcrit_pl)<-c("subj","item","rt","int","x","expt")
wagersE3pl<-E3postcrit_pl

##singular:
E3postcrit_sg$x<-ifelse(E3postcrit_sg$Condition=="c",-1,1)
E3postcrit_sg$int<-ifelse(E3postcrit_sg$Condition=="c","low","high")
E3postcrit_sg<-E3postcrit_sg[,c(4,3,8,15,14)]
E3postcrit_sg$expt<-factor("wagersE3sg")
colnames(E3postcrit_sg)<-c("subj","item","rt","int","x","expt")
wagersE3sg<-E3postcrit_sg

##E4
E4postcrit<-subset(Experiment4,Region==8) ##
```

```
nsubj_wagerse4<-length(unique(E4postcrit$Subj))
#head(subset(Experiment4,Condition=="c"),n=10)
#postcritical region
#E4postcrit$intr.au<-ifelse(E4postcrit$gramm=="ungram" & E4postcrit$match=="match",-1/2,
#                           ifelse(E4postcrit$gramm=="ungram" & E4postcrit$match=="mismatch",1/2,0))
E4postcrit<-subset(E4postcrit,gramm=="ungram")
E4postcrit$Condition<-factor(E4postcrit$Condition)
E4postcrit$x<-ifelse(E4postcrit$Condition=="c",-1,1)
E4postcrit$int<-ifelse(E4postcrit$Condition=="c","low","high")
E4postcrit<-E4postcrit[,c(4,3,8,13,12)]
E4postcrit$expt<-factor("wagersE4")
colnames(E4postcrit)<-c("subj","item","rt","int","x","expt")
wagersE4<-E4postcrit

# E5
E5postcrit<-subset(Experiment5,Region==8) ##postcritical region
nsubj_wagerse5<-length(unique(E5postcrit$Subj))
E5postcrit<-subset(E5postcrit,gramm=="ungram")
E5postcrit$Condition<-factor(E5postcrit$Condition)
## c,d
E5postcrit$x<-ifelse(E5postcrit$Condition=="c",-1,1)
E5postcrit$int<-ifelse(E5postcrit$Condition=="c","low","high")
E5postcrit<-E5postcrit[,c(4,3,8,13,12)]
colnames(E5postcrit)<-c("subj","item","rt","int","x")
E5postcrit$expt<-factor("wagersE5")
wagersE5<-E5postcrit

#head(wagersE5)
```

Assemble all the data into one data frame:

```
dat<-rbind(dillonE1,wagersE2,
           lagoE1,lagoE2,
           lagoE3a,lagoE3b,
           wagersE2,
           wagersE3pl,wagersE3sg,
           wagersE4,wagersE5)
dat$subj<-factor(paste(dat$expt,dat$subj,sep=""))
dat$item<-factor(paste(dat$expt,dat$item,sep=""))
#with(dat,tapply(subj,expt,function(x)length(unique(x))))
head(dat)
```

```
##          subj       item   rt  int  x     expt
## 77  dillonE11 dillonE119  195  low -1 dillonE1
## 203 dillonE11 dillonE135  479  low -1 dillonE1
## 539 dillonE11 dillonE143 2287  low -1 dillonE1
## 665 dillonE11  dillonE14  730 high  1 dillonE1
## 749 dillonE11 dillonE128 1246 high  1 dillonE1
## 791 dillonE11 dillonE127 4027  low -1 dillonE1
```

In the dat data frame above, int is the low/high interference condition, which is +/-1 coded in the column x.

Note that if one uses all the data altogether one has to rename the subjects, because subject 1 in dillonE1 is not the same subject as in some other experiment with subject id 1.

This Stan+lmer code is not run here (results are precomputed and stored)

One can fit the Stan/lmer models to these data using these commands. The results needed in this file are stored in the data directory through these commands.

```
##----agrmtattrn2,echo=FALSE,warning=FALSE,message=FALSE,eval=FALSE-------
#Dillon E1:
stanDat<-createStanDat(d=dillonE1,
                       rt=dillonE1$rt,
                       form=as.formula("~ 1 + x"))
#str(stanDat)
DillonE1 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
             data = stanDat,
             iter = 2000,
             chains = 4)
#Int is Interference:
pars<-c("Int","beta[2]","sigma_u[1]","sigma_u[2]","sigma_w[1]","sigma_w[2]","sigma_e")
DillonE1_res<-stan_results(DillonE1,params=pars[1])

#with(dillonE1,tapply(rt,x,mean))

mDillonE1_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),dillonE1,
        control=lmerControl(calc.derivs=FALSE))
mDillonE1_lmer_res<-summary(mDillonE1_lmer)$coefficients[2,]

stanDat<-createStanDat(d=lagoE1,
                       rt=lagoE1$rt,
                       form=as.formula("~ 1 + x"))

LagoE1 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                 data = stanDat,
                 iter = 2000,
                 chains = 4)
LagoE1_res<-stan_results(LagoE1,params=pars[1])

mLagoE1_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),lagoE1,
        control=lmerControl(calc.derivs=FALSE))
mLagoE1_lmer_res<-summary(mLagoE1_lmer)$coefficients[2,]


stanDat<-createStanDat(d=lagoE2,
                       rt=lagoE2$rt,
                       form=as.formula("~ 1 + x"))

LagoE2 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
               data = stanDat,
               iter = 2000,
               chains = 4)
LagoE2_res<-stan_results(LagoE2,params=pars[1])

mLagoE2_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),lagoE2,
        control=lmerControl(calc.derivs=FALSE))
mLagoE2_lmer_res<-summary(mLagoE2_lmer)$coefficients[2,]
```

```
stanDat<-createStanDat(d=lagoE3a,
                          rt=lagoE3a$rt,
                       form=as.formula("~ 1 + x"))

LagoE3a <- stan(file = "StanModels/maxModelTargetMismatch.stan",
               data = stanDat,
               iter = 2000,
               chains = 4)

LagoE3a_res<-stan_results(LagoE3a,params=pars[1])

mLagoE3a_lmer<-lmer(log(rt)~x+(1|subj)+(1|item),lagoE3a,
        control=lmerControl(calc.derivs=FALSE))
mLagoE3a_lmer_res<-summary(mLagoE3a_lmer)$coefficients[2,]

stanDat<-createStanDat(d=lagoE3b,
                          rt=lagoE3b$rt,
                       form=as.formula("~ 1 + x"))

LagoE3b <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                data = stanDat,
                iter = 2000,
                chains = 4)

LagoE3b_res<-stan_results(LagoE3b,params=pars[1])

mLagoE3b_lmer<-lmer(log(rt)~x+(1+x||subj)+(1|item),lagoE3b,
        control=lmerControl(calc.derivs=FALSE))
mLagoE3b_lmer_res<-summary(mLagoE3b_lmer)$coefficients[2,]


stanDat<-createStanDat(d=wagersE2,
                          rt=wagersE2$rt,
                       form=as.formula("~ 1 + x"))

WagersE2 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                data = stanDat,
                iter = 2000,
                chains = 4)
WagersE2_res<-stan_results(WagersE2,params=pars[1])

mWagersE2_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),wagersE2,
        control=lmerControl(calc.derivs=FALSE))
mWagersE2_lmer_res<-summary(mWagersE2_lmer)$coefficients[2,]

stanDat<-createStanDat(d=wagersE3pl,
                          rt=wagersE3pl$rt,
                       form=as.formula("~ 1 + x"))

WagersE3pl <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                 data = stanDat,
                 iter = 2000,
                 chains = 4)
```

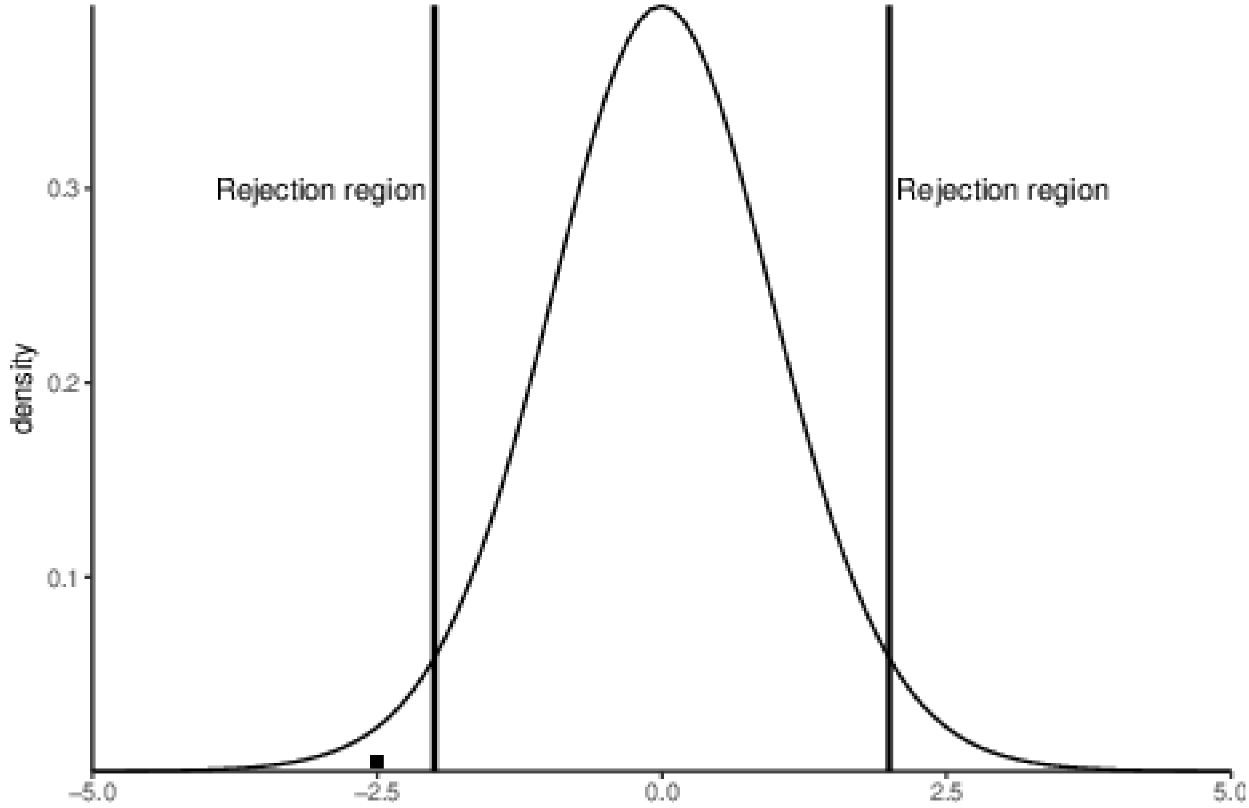
```
WagersE3pl_res<-stan_results(WagersE3pl,params=pars[1])

mWagersE3pl_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),wagersE3pl,
        control=lmerControl(calc.derivs=FALSE))
mWagersE3pl_lmer_res<-summary(mWagersE3pl_lmer)$coefficients[2,]

stanDat<-createStanDat(d=wagersE3sg,
                          rt=wagersE3sg$rt,
                          form=as.formula("~ 1 + x"))

WagersE3sg <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                   data = stanDat,
                   iter = 2000,
                   chains = 4)
WagersE3sg_res<-stan_results(WagersE3sg,params=pars[1])

mWagersE3sg_lmer<-lmer(log(rt)~x+(1+x||subj)+(1+x||item),wagersE3sg,
        control=lmerControl(calc.derivs=FALSE))
mWagersE3sg_lmer_res<-summary(mWagersE3sg_lmer)$coefficients[2,]


stanDat<-createStanDat(d=wagersE4,
                          rt=wagersE4$rt,
                          form=as.formula("~ 1 + x"))

WagersE4 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                   data = stanDat,
                   iter = 2000,
                   chains = 4)
WagersE4_res<-stan_results(WagersE4,params=pars[1])

mWagersE4_lmer<-lmer(log(rt)~x+(1|subj),wagersE4,
        control=lmerControl(calc.derivs=FALSE))
mWagersE4_lmer_res<-summary(mWagersE4_lmer)$coefficients[2,]

stanDat<-createStanDat(d=wagersE5,
                       rt=wagersE5$rt,
                       form=as.formula("~ 1 + x"))

WagersE5 <- stan(file = "StanModels/maxModelTargetMismatch.stan",
                 data = stanDat,
                 iter = 2000,
                 chains = 4)
WagersE5_res<-stan_results(WagersE5,params=pars[1])

mWagersE5_lmer<-lmer(log(rt)~x+(1+x|subj)+(1+x|item),wagersE5,
        control=lmerControl(calc.derivs=FALSE))
mWagersE5_lmer_res<-summary(mWagersE5_lmer)$coefficients[2,]

#posterior predicted data from Jaeger et al replication: See MethodsX paper:
load("../model/au_predicted_meansD13rep.Rda")
len_model<-length(au_predicted_means_rep)
```

```
agrmt_data<-data.frame(expt=factor(rep(c(1:10),
                                         each=4000)),
                       posterior=c(DillonE1_res[2][[1]][[1]],
                       LagoE1_res[2][[1]][[1]],
                       LagoE2_res[2][[1]][[1]],
                       LagoE3a_res[2][[1]][[1]],
                       LagoE3b_res[2][[1]][[1]],
                       WagersE2_res[2][[1]][[1]],
                       WagersE3pl_res[2][[1]][[1]],
                       WagersE3sg_res[2][[1]][[1]],
                       WagersE4_res[2][[1]][[1]],
                       WagersE5_res[2][[1]][[1]]))

means<-sort(with(agrmt_data,tapply(posterior,expt,mean)))
ordered_studies<-as.numeric(names(means))

model_pred<-data.frame(expt=rep("model",len_model),posterior=au_predicted_means_rep)
head(model_pred)

data_model<-rbind(agrmt_data,model_pred)

lvls<-levels(data_model$expt)
data_model$expt<-factor(data_model$expt,levels=lvls[c(11,ordered_studies
)])

save(data_model,file="../data/data_model.Rda")

#frequentist CIs:
lmer_estimates<-data.frame(rbind(mDillonE1_lmer_res,
mLagoE1_lmer_res,
mLagoE2_lmer_res,
mLagoE3a_lmer_res,
mLagoE3b_lmer_res,
mWagersE2_lmer_res,
mWagersE3pl_lmer_res,
mWagersE3sg_lmer_res,
mWagersE4_lmer_res,
mWagersE5_lmer_res))
lmer_estimates$lower<-lmer_estimates[,1]-2*lmer_estimates[,2]
lmer_estimates$upper<-lmer_estimates[,1]+2*lmer_estimates[,2]
lmer_estimates$id<-1:10
lmer_estimates2<-lmer_estimates[order(lmer_estimates[,1]), ]
lmer_estimates3<-lmer_estimates2[,c(6,1,4,5,3)]
save(lmer_estimates2,file="../data/lmer_estimates2.Rda")

save(lmer_estimates3,file="../data/lmer_estimates3.Rda")
```

t-test demo

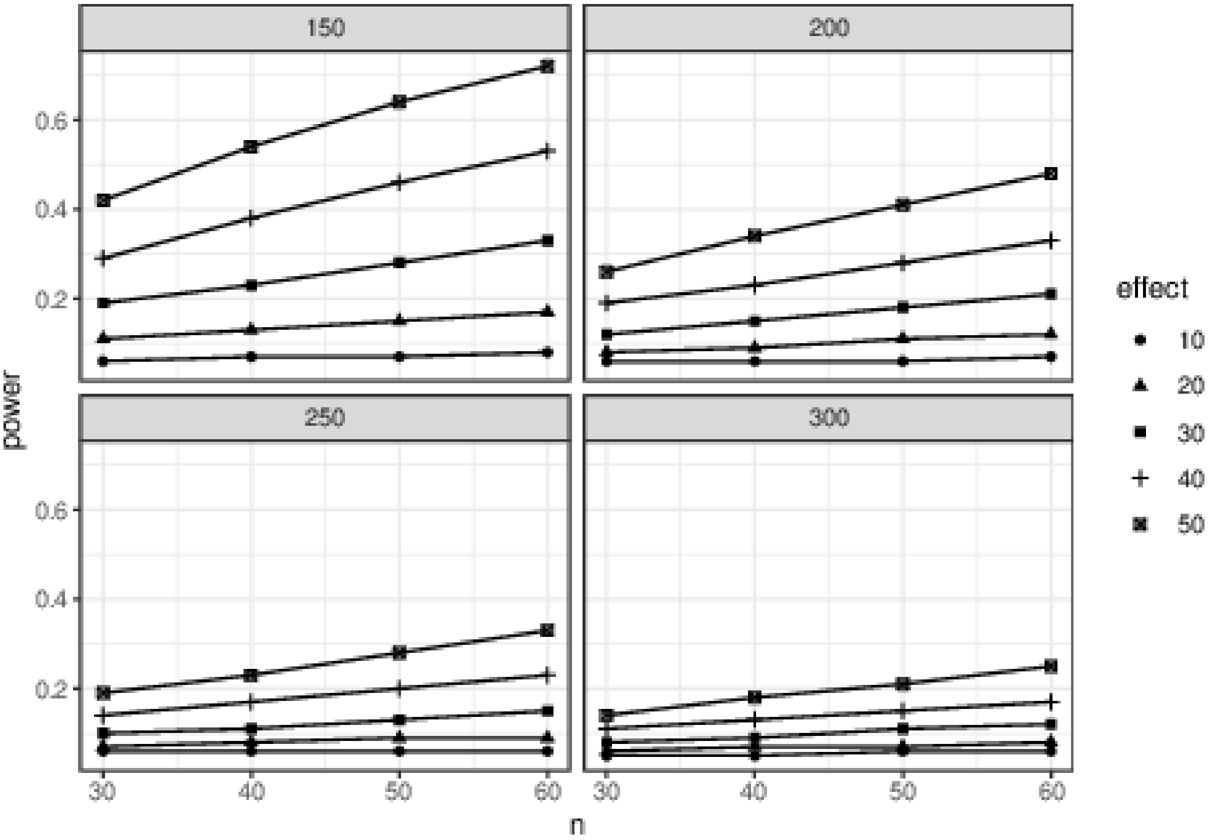
```
## ----ttestdemo,echo=FALSE,fig.height=3-----------------------------------
obst<- -2.5
xend<-5
```

```
degf<-20
x<-seq(-xend,xend,by=0.1)
y<-dt(x,df=degf)
dat<-data.frame(x=x,y=y)

p<-ggplot(data = dat,
          mapping = aes(x = x, y = y)) +
    geom_line()+
    scale_x_continuous(expand = c(0, 0))+
   scale_y_continuous(expand = c(0, 0))+
   geom_vline(xintercept = -2,
                color = "black", size=1)+
   geom_vline(xintercept = 2,
                color = "black", size=1)+
   geom_point(aes(obst,0.005))+
  ylab("density")+xlab("")+theme_bw()+
  annotate("text", x = -3, y = 0.3, label = "Rejection region")+
  annotate("text", x = 3, y = 0.3, label = "Rejection region")

p + theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

# Power
```



```
## ----powerplotfunctions,echo=FALSE,eval=TRUE----------------------------
plot_power<-function(distrn=NULL,
                     xlabl="",
                     ylabl="",
                     titl="",
                     scl=1){
  ggplot(distrn,
       aes(x = power, y = nsubj,
           height = ..density..)) +
  geom_density_ridges(scale = scl,
                      stat = "density", rel_min_height = 0.01) +
  scale_y_discrete(expand = c(0.01, 0)) +
  xlim(0.05,1)+
  theme_ridges() + theme(legend.position = "none")+
  xlab(xlabl)+
  ggtitle(titl)+
  ylab(ylabl)+
  magnifytext(sze=12)
}

simulate_power<-function(nsim=10000,
                         stddev=NULL,
                         nsamps=seq(30,60,by=10),
                         d_mean=30,
                         d_sd=10){
##take nsim samples
##from "prior" on effect:
  d<-rnorm(nsim,mean=d_mean,sd=d_sd)
  powmatrix<-matrix(rep(NA,
                        nsim*length(nsamps)),ncol=length(nsamps))
for(i in 1:nsim){
  powmatrix[i,]<-power.t.test(delta=d[i],
                        n=nsamps,
                        sd=stddev,
                        type="one.sample",strict=TRUE)$power
}
power_df<-data.frame(nsubj=factor(rep(nsamps,
each=nsim)),
           stddev=rep(stddev,
           length(nsamps)*nsim),
power=as.vector(powmatrix))
power_df
}

## ----plotpower,echo=FALSE,fig.width=4,fig.height=5-----------------------
simulate_power2<-function(nsamps=30,
                         d_mean=30,
                         d_sd=NULL,
                         nsim=100){
    pow<-power.t.test(delta=d_mean,
                                n=nsamps,
                                sd=d_sd,
                                type="one.sample",strict=TRUE)$power
    pow
}
```

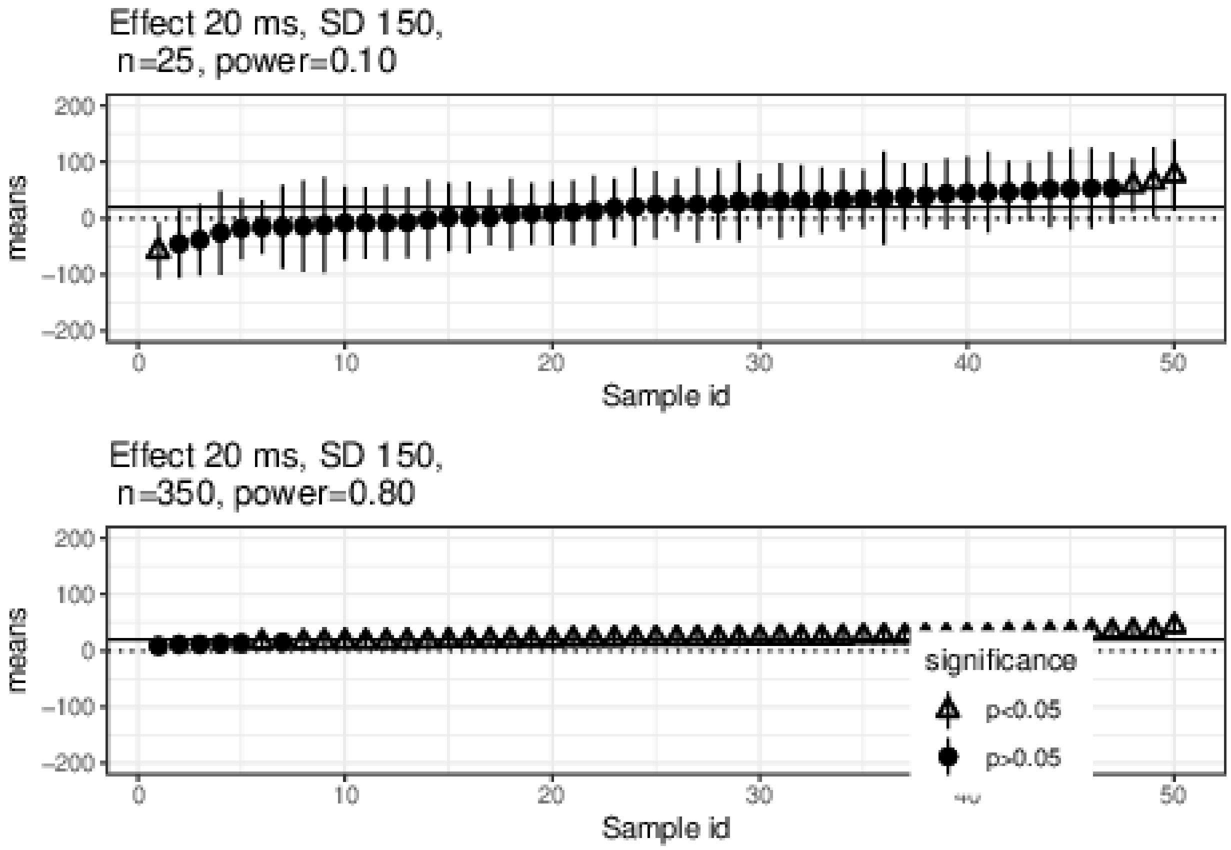
```
samps<-c(30,40,50,60)
sds<-c(150,200,250,300)
ds<-c(10,20,30,40,50)

powvals<-c()

## for each sample size:
for(i in 1:length(samps)){
  ## for each sd:
  for(j in 1:length(sds)){
    ## for each effect size:
    for(k in 1:length(ds)){
pwr<-simulate_power2(nsamps=samps[i],
                d_mean=ds[k],
                d_sd=sds[j])
powvals<-c(powvals,round(pwr,2))
    }
  }
}

power_calc<-data.frame(n=rep(samps,each=5*4),
                       sd=rep(rep(sds,each=5),4),
                       effect=rep(ds,16))

power_calc$powvals<-powvals

power_calc$sd<-factor(power_calc$sd)
power_calc$effect<-factor(power_calc$effect)

power_plot<-ggplot(power_calc, aes(x=n, y=powvals,
                       shape=effect)) + ylab("power")+geom_point()+geom_line()+theme_bw()+facet_wrap(~s

power_plot
```



Type M error demo

```
## ----demotypeM,echo=FALSE,fig.width=7,fig.height=6,include=TRUE----------
set.seed(987654321)
d<-20
sd<-150
lown<-power.t.test(d=d,sd=sd,power=.10,type="one.sample",alternative="two.sided",strict=TRUE)$n
highn<-power.t.test(d=d,sd=sd,power=.80,type="one.sample",alternative="two.sided",strict=TRUE)$n
nsim<-50
tlow<-thigh<-meanslow<-meanshigh<-CIuplow<-CIlwlow<-CIuphigh<-CIlwhigh<-NULL
critlow<-abs(qt(0.025,df=lown-1))
crithigh<-abs(qt(0.025,df=highn-1))

for(i in 1:nsim){
  x<-rnorm(lown,mean=d,sd=sd)
  meanslow[i]<-mean(x)
  tlow[i]<-t.test(x)$statistic
  CIuplow[i]<-mean(x)+critlow*sd(x)/sqrt(length(x))
  CIlwlow[i]<-mean(x)-critlow*sd(x)/sqrt(length(x))
  x<-rnorm(highn,mean=d,sd=sd)
  meanshigh[i]<-mean(x)
  thigh[i]<-t.test(x)$statistic
  CIuphigh[i]<-mean(x)+crithigh*sd(x)/sqrt(length(x))
  CIlwhigh[i]<-mean(x)-crithigh*sd(x)/sqrt(length(x))
}
```

```
siglow<-ifelse(abs(tlow)>abs(critlow),"p<0.05","p>0.05")
sighigh<-ifelse(abs(thigh)>abs(crithigh),"p<0.05","p>0.05")

summarylow<-data.frame(means=meanslow,significance=siglow, CIupper=CIuplow, CIlower=CIlwlow)
summaryhigh<-data.frame(index=1:nsim,means=meanshigh,significance=sighigh, CIupper=CIuphigh, CIlower=CI


# re-order data by mean effect size
summarylow<-summarylow[order(summarylow$means), ]
summarylow$index<-1:nrow(summarylow)
summaryhigh<-summaryhigh[order(summaryhigh$means), ]
summaryhigh$index<-1:nrow(summaryhigh)

p_low<-ggplot(summarylow, aes(y=means, x=index,
                              shape=significance,
                              ymax=CIupper, ymin=CIlower)) +
  geom_pointrange()+
  #coord_flip()+
  geom_point(size=2.5)+
  scale_shape_manual(values=c(2, 19))+
  magnifytext(sze=22)+
  geom_hline(yintercept=20) +
  theme_bw() +
  scale_x_continuous(name = "Sample id")+
  scale_y_continuous(name = "means",limits=c(-200,200))+
  labs(title="Effect 20 ms, SD 150, \n n=25, power=0.10")+
  theme(legend.position="none")+geom_hline(yintercept=0, linetype="dotted")

p_hi<-ggplot(summaryhigh, aes(y=means, x=index,
                              shape=significance, ymax=CIupper, ymin=CIlower)) +
  geom_pointrange()+
  #coord_flip()+
  geom_point(size=2.5)+
  scale_shape_manual(values=c(2, 19))+
    scale_x_continuous(name = "Sample id")+
  magnifytext(sze=22)+
  geom_hline(yintercept=d) +
  theme_bw() +
  scale_y_continuous(name = "means",limits=c(-200,200))+
  labs(title="Effect 20 ms, SD 150, \n n=350, power=0.80")+
  theme(legend.position=c(0.8,0.25))+geom_hline(yintercept=0, linetype="dotted")

multiplot(p_low,p_hi,cols=1)
```



t-values across the studies

```r
## ----echo=FALSE,results=`asis'--------------------------------------------
load("../data/tvals.Rda")
rownames(tvals)<-""
xtable(round(tvals,2),caption="t-values from 10 published studies on the agreement attraction effect.",
```

% latex table generated in R 4.1.2 by xtable 1.8-4 package % Sat Mar 19 22:07:25 2022

1	2	3	4	5	6	7	8	9	10
-2.56	-2.25	-1.67	-1.83	-1.40	-2.22	-1.33	-0.22	-2.81	-1.74

Table 1: t-values from 10 published studies on the agreement attraction effect.

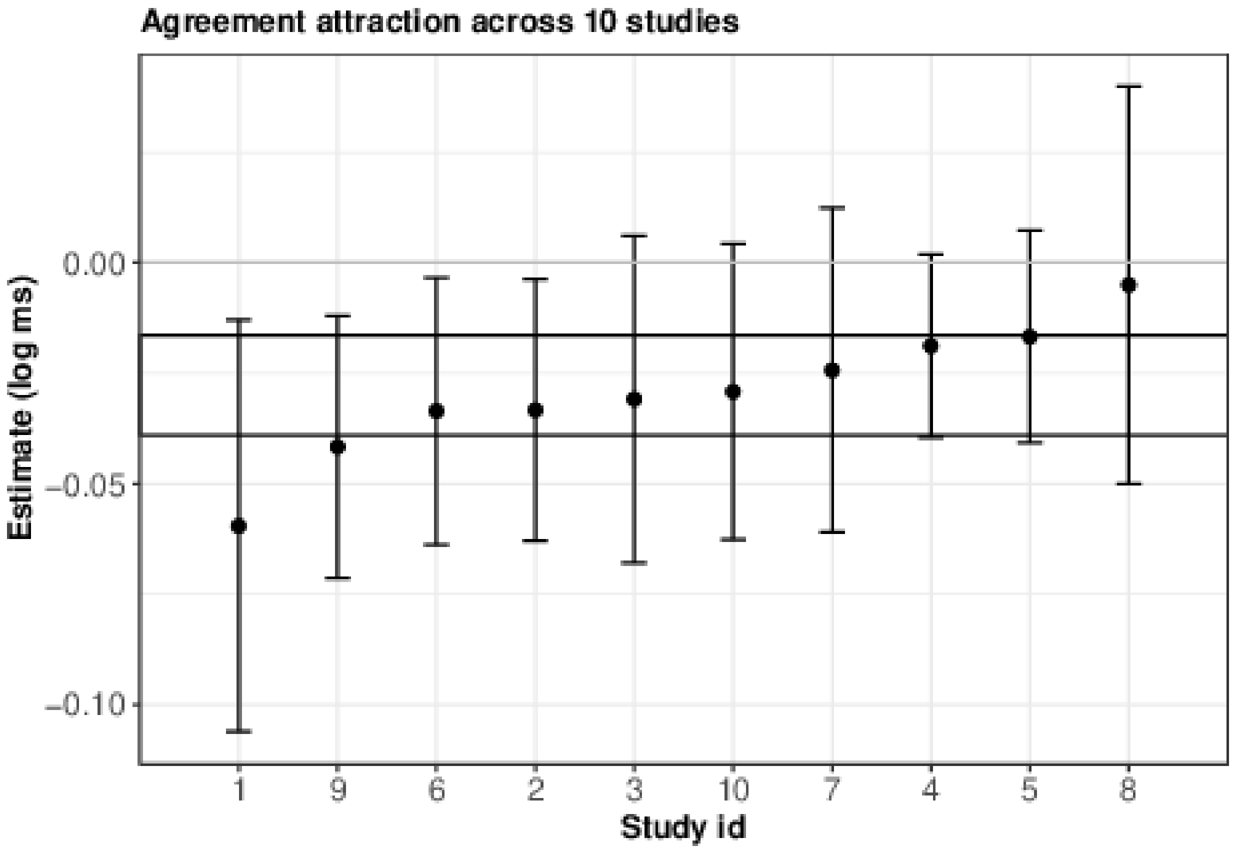
Meta-analysis

Given the data from different studies, one can do the meta-analysis by running the following commented out code. Please see https://github.com/vasishth/MetaAnalysisJaegerEngelmannVasishth2017 for more.

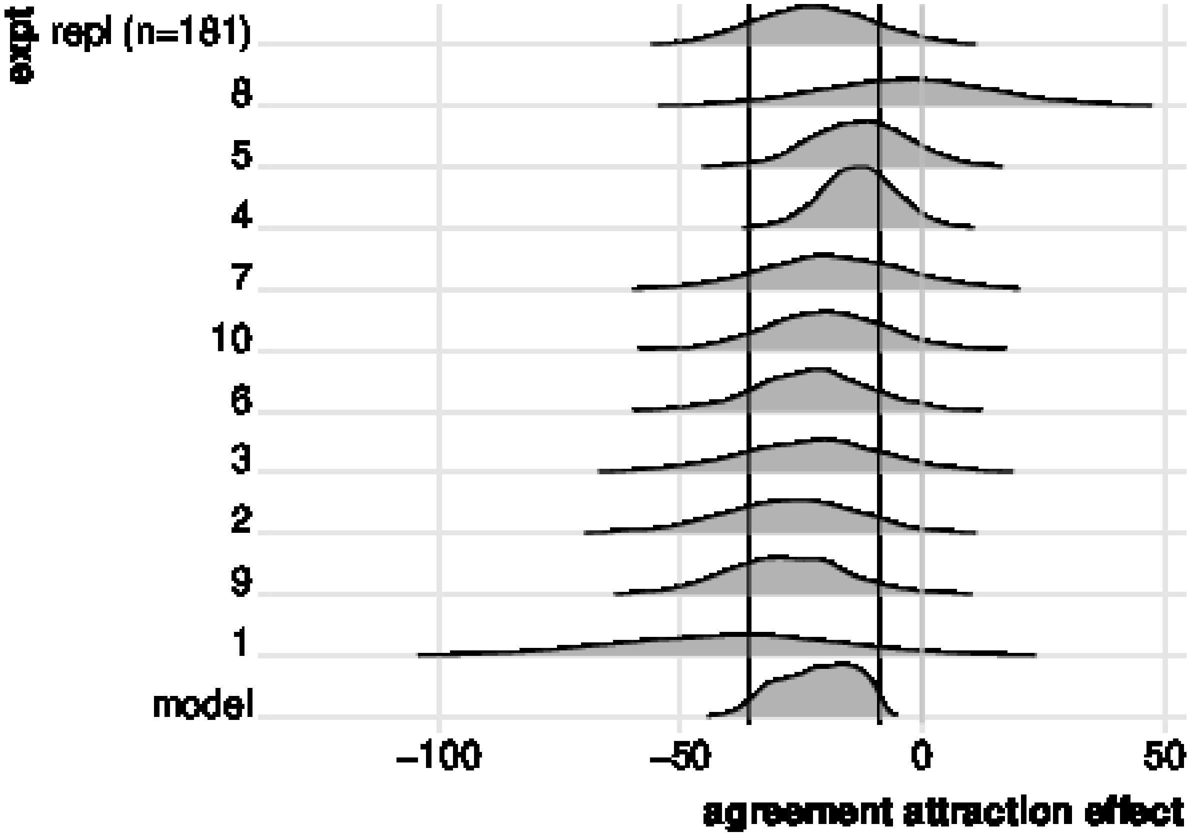
```r
## ----metaanalysis,echo=FALSE,warning=FALSE,message=FALSE,eval=FALSE------
## load("../data/lmer_estimates2.Rda")
## dat<-list(N=dim(lmer_estimates2)[1],
##           y=lmer_estimates2$Estimate,
##           sigma=lmer_estimates2$Std..Error)
##
## fit <- stan(file='../StanModels/rema.stan', data=dat,
##             iter=2000, chains=4, seed=987654321,
##             control = list(adapt_delta = 0.8))
## save(fit,file="../data/remafit.Rda")
```

Load pre-computed meta-analysis values:

```
## ----remafitload,echo=FALSE----------------------------------------------
load("../data/remafit.Rda")
paramnames<-c("theta","tau")
#print(fit,pars=paramnames)

params<-extract(fit,pars=paramnames)
#str(params)

mean_theta<-mean(params$theta)
lower_theta<-quantile(params$theta,0.025)
upper_theta<-quantile(params$theta,0.975)


## ----dillonreprise,echo=FALSE--------------------------------------------
mean_dillonma<-round(exp(6.4376+mean_theta)-exp(6.4376-mean_theta))
upper_dillonma<-round(exp(6.4376+(upper_theta))-exp(6.4376-(upper_theta)))
lower_dillonma<-round(exp(6.4376+(lower_theta))-exp(6.4376-(lower_theta)))


## ----echo=FALSE,fig.width=5,fig.height=4---------------------------------
load("../data/lmer_estimates3.Rda")
lmer_estimates3$id<-factor(lmer_estimates3$id,levels=lmer_estimates3$id)
pd<-position_dodge(0.6)
ggplot(lmer_estimates3, aes(x=id,
                               y=Estimate,group=id)) +
    geom_errorbar(aes(ymin=lower, ymax=upper),
                  width=.25, size=.5, position=pd) +
      annotate("rect",
             xmin = 0,
             xmax = 11,
             ymin = upper_theta,
             ymax = lower_theta,
             color = "black",alpha=0.2)+
#    geom_hline(yintercept=mean_theta,
#               color="black",)+
    labs(title="Agreement attraction across 10 studies") +
    xlab("Study id")+
    ylab("Estimate (log ms)")+
    geom_hline(yintercept=0,col="gray")+
    geom_point(position=pd, size=2)+
    theme_bw()+
    magnifytext()
```



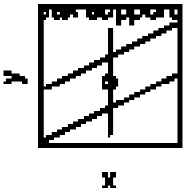
```
## ----ridgeplot,echo=FALSE,fig.width=9,fig.height=10---------------------
## precomputed above:
load("../data/data_model.Rda")
load("../data/data_model_dillonrep.Rda")
modelquantiles<-quantile(subset(data_model,expt=="model")$posterior,prob=c(0.025,0.975))

expt_dillonrep<-subset(data_model_dillonrep,expt==11)
#head(expt_dillonrep)
expt_dillonrep$expt<-factor("repl (n=181)")
data_model11studies<-rbind(data_model,expt_dillonrep)

scl<-1
ggplot(data_model11studies,
       aes(x = posterior, y = factor(expt),height = ..density..
           )) +
  geom_density_ridges(scale = scl
                      ,stat = "density",
                      rel_min_height = 0.01) +
  scale_y_discrete(expand = c(0.01, 0)) +
  scale_x_continuous(expand = c(0.01, 0)) +
  #scale_fill_brewer(palette = "PuBuGn") +
  theme_ridges() + theme(legend.position = "none")+
  xlab("agreement attraction effect")+
  ylab("expt")+
  geom_vline(xintercept=0,col="gray")+
  ## meta-analysis based on frequentist estimates
  geom_vline(xintercept=-9)+
  geom_vline(xintercept=-36)+
    magnifytext(sze=14)
```

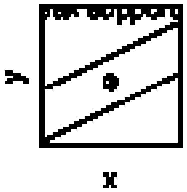


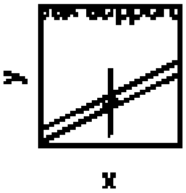
A graphical summary of the Roberts and Pashler 2000 criteria

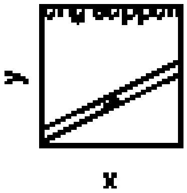
```
## ----rp,echo=FALSE,fig.height=6,fig.width=6-----------------------------
y<-seq(1,100,by=0.01)

op<-par(mfrow=c(2,2),pty="s")

plot(y,y,type="l",ylim=c(0,200),
     main="",xaxt="n",yaxt="n",xlab="",ylab="")
title(xlab="x",
      line=.5, cex.lab=1)
title(ylab="y",
      line=.5, cex.lab=1)

lines(y,y+80)
points(50,90)
arrows(x0=50,x1=50,y0=10,y1=170,
       angle=90,
       length=0)
text("weak support",x=40,y=200)

plot(y,y,type="l",ylim=c(0,200),
     main="",xaxt="n",yaxt="n",xlab="",ylab="")
```

```
title(xlab="x",
      line=.5, cex.lab=1)
title(ylab="y",
      line=.5, cex.lab=1)
lines(y,y+80)
points(50,90)
arrows(x0=50,x1=50,y0=80,y1=100,
       angle=90,
       length=0)
text("weak support",x=40,y=200)

plot(y,y,type="l",ylim=c(0,200),
     main="",xaxt="n",yaxt="n",xlab="",ylab="")
title(xlab="x",
      line=.5, cex.lab=1)
title(ylab="y",
      line=.5, cex.lab=1)
lines(y,y+20)
points(50,60)
arrows(x0=50,x1=50,y0=60-50,y1=60+50,
       angle=90,
       length=0)
text("weak support",x=40,y=200)

plot(y,y,type="l",ylim=c(0,200),
     main="",xaxt="n",yaxt="n",xlab="",ylab="")
title(xlab="x",
      line=.5, cex.lab=1)
title(ylab="y",
      line=.5, cex.lab=1)
lines(y,y+20)
points(50,60)
arrows(x0=50,x1=50,y0=60-10,y1=60+10,
       angle=90,
       length=0)
text("strong support",x=40,y=200)
```









Computing power using simulation

```
## ----computepower,echo=FALSE-----------------------------------------------
source("../R/gen_fake_norm.R")

## maximal model, ignore correlations:
mDillonE1_lmer<-lmer(log(rt)~x+(1+x|subj)+(1+x|item),dillonE1,
        control=lmerControl(calc.derivs=FALSE))

extract_parests_lmer<-function(
  mod=mDillonE1_lmer){
  alpha<-summary(mod)$coefficients[1,1]
  beta<-summary(mod)$coefficients[2,1]
## extract standard deviation estimate:
sigma_e<-attr(VarCorr(mod),"sc")
## assemble variance covariance matrix for subjects:
subj_ranefsd<-attr(VarCorr(mod)$subj,"stddev")
sigma_u0<-subj_ranefsd[1]
sigma_u1<-subj_ranefsd[2]
item_ranefsd<-attr(VarCorr(mod)$item,"stddev")
sigma_w0<-item_ranefsd[1]
sigma_w1<-item_ranefsd[2]
## return list of params:
list(alpha=alpha,beta=beta,sigma_e=sigma_e,sigma_u0=sigma_u0,sigma_u1=sigma_u1,sigma_w0=sigma_w0,sigma_
}

parest<-extract_parests_lmer()

compute_power<-function(nsim=100,
                        alpha=parest$alpha,
                        beta=parest$beta,
                        sigma_e=parest$sigma_e,
                        sigma_u0=parest$sigma_u0,
                        sigma_u1=parest$sigma_u1,
                sigma_w0=parest$sigma_w0,
                        sigma_w1=parest$sigma_w1,
                        nsubj=48,
                        nitem=40){
tvals<-c()
for(i in 1:nsim){
fakedat<-gen_fake_norm(nitem=nitem,
                       nsubj=nsubj,
               alpha=alpha,
               beta=beta,
               sigma_u0=sigma_u0,
               sigma_u1=sigma_u1,
               sigma_w0=sigma_w0,
               sigma_w1=sigma_w1,
               sigma_e=sigma_e)
m<-lmer(rt~cond+(1+cond||subj)+(1+cond||item),
        fakedat,
        control=lmerControl(calc.derivs=FALSE))
tvals[i]<-summary(m)$coefficients[2,3]
}
mean(abs(tvals)>2)
}
```

This code will take some time to run the first time round, but after that the results will be cached. If you change the code below in any way, I suggest deleting the cache directory and recomputing.

```
## ----powercalcstudy1,echo=FALSE,cache=TRUE------------------------------
pow_meanma<-round(compute_power(beta=mean_theta),2)
## lower theta is the bigger value in absolute terms
## so it gives the upper bound of power:
pow_upperma<-round(compute_power(beta=lower_theta),2)
pow_lowerma<-round(compute_power(beta=upper_theta),2)
#save(pow_meanma,"../data/pow_meanma.Rda")
#save(pow_upperma,"../data/pow_upperma.Rda")
#save(pow_lowerma,"../data/pow_lowerma.Rda")


## ----powercalcstudy1_300,echo=FALSE,cache=TRUE--------------------------
pow_meanma_300<-round(compute_power(beta=mean_theta,nsubj=300),2)
## lower theta is the bigger value in absolute terms
## so it gives the upper bound of power:
pow_upperma_300<-round(compute_power(beta=lower_theta,nsubj=300),2)
pow_lowerma_300<-round(compute_power(beta=upper_theta,nsubj=300),2)
```

Chinese RC example (power analysis):

```
## ----chineseRCexample,echo=FALSE-----------------------------------------
gw<-read.table("../data/gibsonwu2012data.txt",
               header=TRUE)
## sum-contrast coding of predictor:
gw$cond <- ifelse(
  gw$type%in%c("subj-ext"),-1/2,1/2)
## subset critical region
```

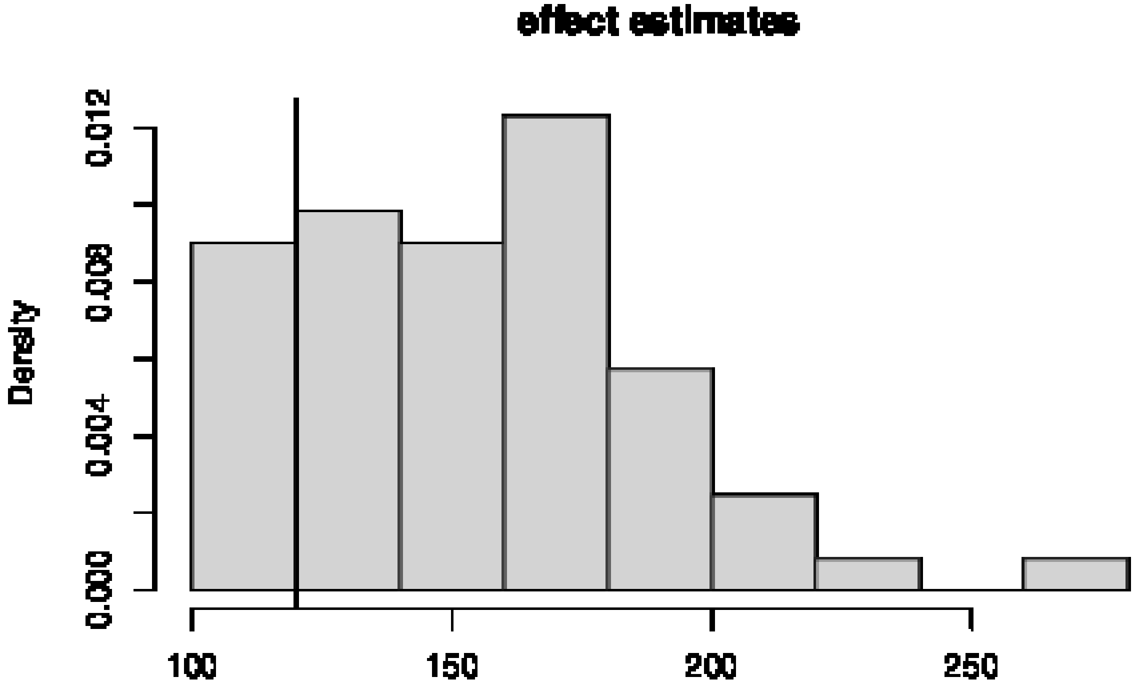
```
dat<-subset(gw,region=="headnoun")


## maximal model; switch off warnings, and ignore the correlations,
## which were not estimable.
m<-lmer(rt~cond+(1+cond|subj)+(1+cond|item),dat,
        control=lmerControl(calc.derivs=FALSE))

## function for extracting all parameter estimates:
extract_parests_lmer<-function(
  mod=m){
  alpha<-summary(mod)$coefficients[1,1]
  beta<-summary(mod)$coefficients[2,1]
## extract standard deviation estimate:
sigma_e<-attr(VarCorr(mod),"sc")
## assemble variance covariance matrix for subjects and items:
subj_ranefsd<-attr(VarCorr(mod)$subj,"stddev")
sigma_u0<-subj_ranefsd[1]
sigma_u1<-subj_ranefsd[2]
item_ranefsd<-attr(VarCorr(mod)$item,"stddev")
sigma_w0<-item_ranefsd[1]
sigma_w1<-item_ranefsd[2]
## return list of params:
list(alpha=alpha,beta=beta,sigma_e=sigma_e,
     sigma_u0=sigma_u0,sigma_u1=sigma_u1,
     sigma_w0=sigma_w0,sigma_w1=sigma_w1)
}

parest<-extract_parests_lmer(mod=m)
```

Demonstrating Type M error

```
compute_typem<-function(nsim=100,
                        alpha=parest$alpha,
                        beta=parest$beta,
                        sigma_e=parest$sigma_e,
                        sigma_u0=parest$sigma_u0,
                        sigma_u1=parest$sigma_u1,
                sigma_w0=parest$sigma_w0,
                        sigma_w1=parest$sigma_w1,
                        nsubj=40,
                        nitem=16){
beta_est<-tvals<-c()
for(i in 1:nsim){
fakedat<-gen_fake_norm(nitem=nitem,
                       nsubj=nsubj,
               alpha=alpha,
               beta=beta,
               sigma_u0=sigma_u0,
               sigma_u1=sigma_u1,
               sigma_w0=sigma_w0,
               sigma_w1=sigma_w1,
               sigma_e=sigma_e)
m<-lmer(rt~cond+(1+cond||subj)+(1+cond||item),
        fakedat,
        control=lmerControl(calc.derivs=FALSE))
tvals[i]<-summary(m)$coefficients[2,3]
beta_est[i]<-summary(m)$coefficients[2,1]
}
beta_est[which(abs(tvals)>2)]
}
```

```
## ----effectestimates,echo=FALSE------------------------------------------
beta_estimates<-compute_typem(beta=120)
propoverestimated<-round(mean(beta_estimates>120),2)
inflation<-beta_estimates/120

beta_estimates60<-compute_typem(beta=60)
propoverestimated60<-round(mean(beta_estimates60>60),2)
inflation60<-beta_estimates60/60
```

```
## ----ploteffectestimates,echo=FALSE,cache=TRUE,fig.height=3,fig.width=3----
hist(beta_estimates,main="effect estimates",xlab="",freq=FALSE)
abline(v=120,lwd=2)
```


effect estimates
Density