```r
rm(list=ls())
options(stringsAsFactors = FALSE)
graphics.off()
setwd("~/Documents/git/proterant/sub_projs/")

library(ape)
library(phytools)

## Loading required package:  maps

library(brms)

## Loading required package:  Rcpp
## Registered S3 method overwritten by 'xts':
##  method    from
##  as.zoo.xts zoo
## Loading 'brms' package (version 2.11.1).  Useful instructions
## can be found by typing help('brms').  A more detailed introduction
## to the package is available through vignette('brms_overview').

library(tibble)
library(ggstance)
library(ggplot2)

##
## Attaching package:  'ggplot2'
## The following objects are masked from 'package:ggstance':
##
##    geom_errorbarh, GeomErrorbarh

library("dplyr")

##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union

library("jpeg")


#useful function
extract_coefs4HF<-function(x){rownames_to_column(as.data.frame(fixef(x, summary=TRUE,probs=c
}
```

```r
##read in the data
HF<-read.csv("HarvardForest/hf003-05-mean-ind.csv",header=TRUE)
HFsubber<-read.csv("HarvardForest/HFdata4modeling.csv",header=TRUE)
HF.tree<-read.tree("HarvardForest/HFtree4modeling.tre")
###make fls measure
HF$phys.fls<-HF$bb.jd-HF$fbb.jd
HF$funct.fls<-HF$l75.jd-HF$fopn.jd
HF$inter.fls<-HF$bb.jd-HF$fopn.jd

###make catagorical FLS
HF$hyst.funct<-ifelse(HF$funct.fls>0,1,0)
HF$hyst.phys<-ifelse(HF$phys.fls>0,1,0)
HF$hyst.inter<-ifelse(HF$inter.fls>0,1,0)

### prune the tree
HF<-dplyr::filter(HF,species!=("QUAL")) ## quercus alba has no flowers
spforcontmods<-HFsubber$species ##subset of species good for this analysis

HF.data<-dplyr::filter(HF,species %in% c(spforcontmods))
HF.data<-dplyr::left_join(HF.data,HFsubber, by="species") ###This is the data for the conti
##zscore predictors for these models
HF.data$pol_cent<-(HF.data$pol-mean(HF.data$pol,na.rm=TRUE))/(2*sd(HF.data$pol,na.rm=TRUE))
HF.data$precip_cent<-(HF.data$min_precip-mean(HF.data$min_precip))/(2*sd(HF.data$min_precip)
HF.data$flo_cent<-(HF.data$fopn.jd-mean(HF.data$fopn.jd,na.rm=TRUE))/(2*sd(HF.data$fopn.jd,r

HF.data$flo_cent.neg<--(HF.data$flo_cent)
HF.data$precip_cent.neg<--(HF.data$precip_cent)
###group by phylogeny
inv.phylo <- MCMCglmm::inverseA(HF.tree, nodes = "TIPS", scale = TRUE)
A <- solve(inv.phylo$Ainv)
rownames(A) <- rownames(inv.phylo$Ainv)


modelcont.funct.wspecies.ind<-brm(funct.fls~ pol+flo_cent+precip_cent+precip_cent:flo_cent+p

## Warning:  Rows containing NAs were excluded from the model.
## Compiling the C++ model
## Start sampling

##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000227 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.27 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
```

```
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 1: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 1: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 1: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 53.4073 seconds (Warm-up)
## Chain 1:                17.1667 seconds (Sampling)
## Chain 1:                70.5741 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000138 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.38 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 2: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 2: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 2: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 2: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 2: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 52.267 seconds (Warm-up)
## Chain 2:                17.3332 seconds (Sampling)
## Chain 2:                69.6002 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 3).
```

```
## Chain 3:
## Chain 3: Gradient evaluation took 0.000135 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 1.35 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 3: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 3: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 3: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 3: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 3: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 51.7829 seconds (Warm-up)
## Chain 3:                17.1044 seconds (Sampling)
## Chain 3:                68.8874 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000231 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 2.31 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 4: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 4: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 4: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 4: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 4: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 51.714 seconds (Warm-up)
```

```
## Chain 4:                15.5456 seconds (Sampling)
## Chain 4:                67.2596 seconds (Total)
## Chain 4:
```

```
meanflo<-HF.data %>% group_by(name) %>% summarise(meanflotime=mean(fopn.jd,na.rm=TRUE))
HF.data<-left_join(HF.data,meanflo)
```

```
## Joining, by = "name"
```

```
HF.data$within_spec_cf <- HF.data$fopn.j -HF.data$meanflotime

HF.data$meanflocent<-(HF.data$meanflotime-mean(HF.data$meanflotime,na.rm=TRUE))/(2*sd(HF.dat
HF.data$varflocent<-(HF.data$within_spec_cf-mean(HF.data$within_spec_cf,na.rm=TRUE))/(2*sd(H
##side bar what if we take the mean of flotime
modelcont.funct.wspecies.ind.proper<-brm(funct.fls~ pol+meanflocent+within_spec_cf+precip_ce
                                  family = gaussian(), cov_ranef = list(name= A),control=lis
```

```
## Warning:  Rows containing NAs were excluded from the model.
## Compiling the C++ model
## recompiling to avoid crashing R session
## Start sampling

##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000357 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 3.57 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 1: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 1: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 1: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 56.7947 seconds (Warm-up)
## Chain 1:                16.531 seconds (Sampling)
## Chain 1:                73.3257 seconds (Total)
## Chain 1:
```

```
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000133 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.33 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 2: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 2: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 2: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 2: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 2: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 53.3353 seconds (Warm-up)
## Chain 2:                16.2208 seconds (Sampling)
## Chain 2:                69.5561 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000329 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 3.29 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 3: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 3: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 3: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 3: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 3: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
```

```
## Chain 3:
## Chain 3:  Elapsed Time: 51.4752 seconds (Warm-up)
## Chain 3:                16.4109 seconds (Sampling)
## Chain 3:                67.8861 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '3725a5ea07c66760c9301397ec859067' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000147 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.47 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 4: Iteration: 2400 / 4000 [ 60%]  (Warmup)
## Chain 4: Iteration: 2800 / 4000 [ 70%]  (Warmup)
## Chain 4: Iteration: 3001 / 4000 [ 75%]  (Sampling)
## Chain 4: Iteration: 3400 / 4000 [ 85%]  (Sampling)
## Chain 4: Iteration: 3800 / 4000 [ 95%]  (Sampling)
## Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 54.2723 seconds (Warm-up)
## Chain 4:                16.4719 seconds (Sampling)
## Chain 4:                70.7442 seconds (Total)
## Chain 4:
```

```
funct.cont<-extract_coefs4HF(modelcont.funct.wspecies.ind)
funct.bin<-extract_coefs4HF(modelcont.funct.wspecies.ind.proper)

cont<-funct.cont
bin<-funct.bin
cont$data_type<-"my way"
bin$data_type<-"proper"


bin<-dplyr::filter(bin,trait!="Intercept")
bin$trait[which(bin$trait=="pol")]<-"pollination syndrome"
bin$trait[which(bin$trait=="flo_cent")]<- "earlier flowering"
bin$trait[which(bin$trait=="precip_cent")]  <- "water dynamics"
bin$trait[which(bin$trait=="pol:precip_cent")]<- "pollination:water dynamics"
bin$trait[which(bin$trait=="pol:flo_cent")]<-"pollination:earlier flowering"
```

```r
bin$trait[which(bin$trait=="flo_cent:precip_cent")]<-"earlier flowering:water dynamics"



cont<-dplyr::filter(cont,trait!="Intercept")
cont$trait[which(cont$trait=="pol")]<-"pollination syndrome"
cont$trait[which(cont$trait=="flo_cent")]<- "earlier flowering"
cont$trait[which(cont$trait=="precip_cent")]  <- "water dynamics"
cont$trait[which(cont$trait=="pol:precip_cent")]<- "pollination:water dynamics"
cont$trait[which(cont$trait=="pol:flo_cent")]<-"pollination:earlier flowering"
cont$trait[which(cont$trait=="flo_cent:precip_cent")]<-"earlier flowering:water dynamics"

summary(modelcont.funct.wspecies.ind)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: funct.fls ~ pol + flo_cent + precip_cent + precip_cent:flo_cent + precip_cent:po
##    Data: HF.data (Number of observations: 679)
## Samples: 4 chains, each with iter = 4000; warmup = 3000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~name (Number of levels: 23)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    11.44      2.47     7.36    17.18 1.00     1522     2417
##
## ~tree.id (Number of levels: 78)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.33      0.80     0.08     2.98 1.01      663     1595
##
## ~tree.id:species (Number of levels: 78)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.38      0.80     0.08     2.96 1.01      656     1454
##
## Population-Level Effects:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                9.70      4.47     0.84    18.55 1.00     1116
## pol                      7.22      5.09    -3.17    17.35 1.00     1682
## flo_cent               -22.68      2.29   -27.07   -18.18 1.00     3351
## precip_cent            -11.78      7.27   -26.72     2.05 1.00     1492
## flo_cent:precip_cent     0.49      2.94    -5.20     6.36 1.00     4908
## pol:precip_cent         13.56      7.63    -0.88    29.40 1.00     1493
## pol:flo_cent            -5.72      2.89   -11.35    -0.11 1.00     3775
##                      Tail_ESS
## Intercept                1966
## pol                      2013
```

```
## flo_cent                 3039
## precip_cent              1715
## flo_cent:precip_cent     3088
## pol:precip_cent          1979
## pol:flo_cent             2959
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     6.64      0.19     6.28     7.04 1.00     4615     2991
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
summary(modelcont.funct.wspecies.ind.proper)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: funct.fls ~ pol + meanflocent + within_spec_cf + precip_cent + precip_cent:meanf
##    Data: HF.data (Number of observations: 679)
## Samples: 4 chains, each with iter = 4000; warmup = 3000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~name (Number of levels: 23)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     8.46      1.88     5.47    12.67 1.00     1696     2582
##
## ~tree.id (Number of levels: 78)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.42      0.82     0.06     3.11 1.01      538     1400
##
## ~tree.id:species (Number of levels: 78)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.35      0.80     0.08     2.97 1.01      558     1479
##
## Population-Level Effects:
##                            Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                     10.56      3.53     3.61    17.61 1.00
## pol                           -2.09      4.98   -11.98     7.34 1.00
## meanflocent                  -31.58      5.03   -41.86   -21.75 1.00
## within_spec_cf                -0.48      0.05    -0.58    -0.37 1.00
## precip_cent                    2.95      7.77   -12.37    19.00 1.00
## meanflocent:precip_cent       -5.44      6.89   -18.62     8.24 1.00
## pol:precip_cent               -0.92      8.78   -18.79    16.05 1.00
## pol:meanflocent              -10.01      8.87   -27.40     7.98 1.00
```

```
## pol:within_spec_cf            -0.16      0.07    -0.29    -0.02 1.00
## within_spec_cf:precip_cent     0.05      0.07    -0.09     0.19 1.00
##                            Bulk_ESS Tail_ESS
## Intercept                      1599     2170
## pol                            1963     2239
## meanflocent                    1753     2163
## within_spec_cf                 4246     3396
## precip_cent                    1581     1605
## meanflocent:precip_cent        1919     2049
## pol:precip_cent                1603     1708
## pol:meanflocent                1904     2620
## pol:within_spec_cf             4235     3274
## within_spec_cf:precip_cent     7286     2817
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     6.63      0.19     6.27     7.00 1.00     4120     2763
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
both<-rbind(cont,bin)

pd=position_dodgev(height=0.4)
both %>%
  arrange(Estimate) %>%
  mutate(trait = factor(trait, levels=c("earlier flowering:water dynamics","pollination:earl
  ggplot(aes(Estimate,trait))+geom_point(aes(shape=data_type),position=pd,size=3,stroke=.5)+
  geom_errorbarh(aes(xmin=Q2.5,xmax=Q97.5,group=data_type),position=pd,height=0,linetype="do
  geom_errorbarh(aes(xmin=Q10,xmax=Q90,group=data_type),position=pd,height=0,linetype="solid
  theme_linedraw(base_size = 11)+geom_vline(aes(xintercept=0),color="black")+
  xlim(-30,30)+scale_color_manual(values=c("firebrick4"))+scale_shape_discrete(name = "data
```

```
## Warning:  Removed 1 rows containing missing values (geom_point).
## Warning:  Removed 1 rows containing missing values (geom_errorbarh).
## Warning:  Removed 1 rows containing missing values (geom_errorbarh).
```