Bayesian analysis of the NESTA study of interventions against verbal aggression online

Rafal Urbaniak

Contents

1 Exploration 1

2 Causal inference 7

3 Model selection 15

4 Inspecting the model and effect sizes 20

5 Direct effect 27

References 36

1 Exploration

Load the dataset and take a look first.

```
summaries <- read.csv(file = "datasets/Summaries.csv")
head(summaries) %>% kable( "latex", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "scale_down") ,font_size = 9)
```

The basic variables we are dealing with are in the following table.

Further variables are defined in terms of those, in particular, we will be predicting AdiffS which is the standardized difference AA-AB, and AdiffS, which is the standardized difference CA-CB. Before we proceed, we will also standardize the predictors, and add a numerical index for the group:

```
summaries$ABS <- standardize(summaries$AB)
summaries$CBS <- standardize(summaries$CB)
summaries$AAS <- standardize(summaries$AA)
summaries$CAS <- standardize(summaries$CA)
summaries$CDS <- standardize(summaries$CD)
summaries$ADS <- standardize(summaries$AD)
summaries$group <- as.factor(summaries$group)
summaries$groupID <-  as.integer( as.factor(summaries$group) )
```

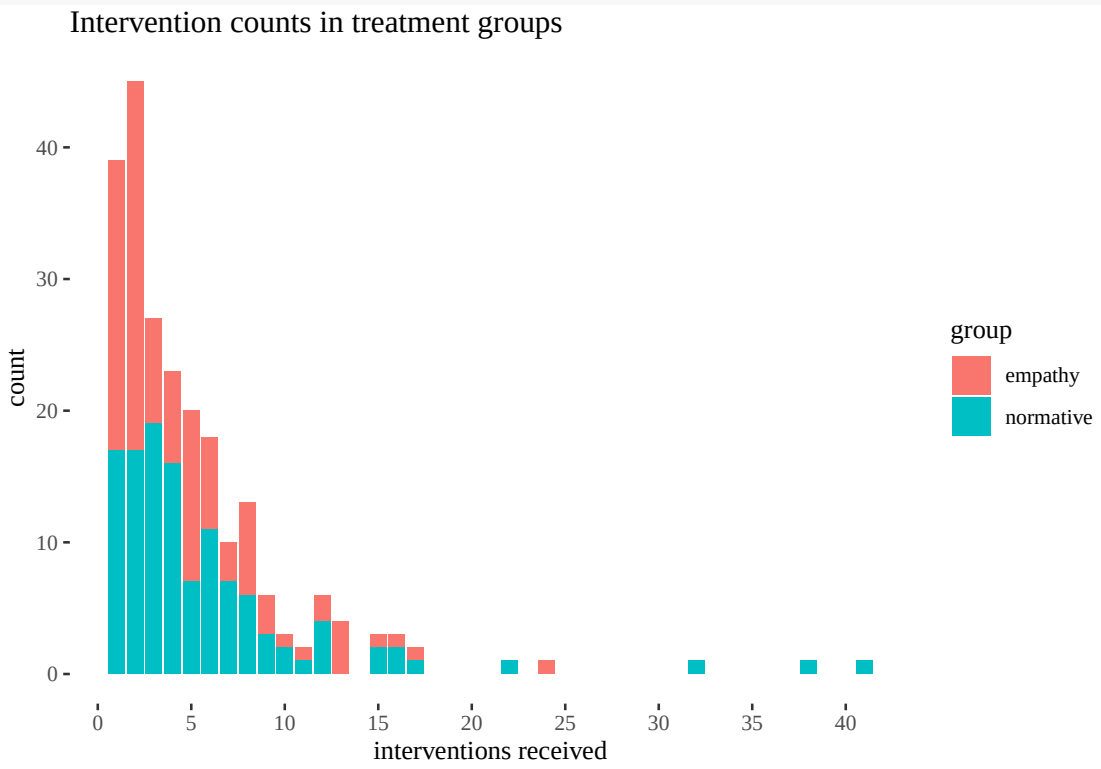
First, let's take a look at the distribution of IC in the treatment groups:

```
ggplot(summaries[summaries$group != "control",], aes(x = IC, fill = group))+
  geom_bar()+theme_tufte()+
  xlab("interventions received")+
```

X	author	AB	AD	AA	CB	CD	CA	Adiff	Cdiff	AdiffS	CdiffS	group	IC
1	_swf	19	1	0	720	25	28	-19	-692	-0.0245122	-0.3501491	normative	1
2	-Allergic	24	24	8	1614	1451	1237	-16	-377	0.0719197	0.1057675	normative	3
3	-funny-username-	23	6	12	847	497	721	-11	-126	0.2326395	0.4690535	control	0
4	-Johnny-	18	2	8	1465	408	684	-10	-781	0.2647835	-0.4789637	empathy	2
5	1secwhileiyeet3	15	3	4	1384	198	120	-11	-1264	0.2326395	-1.1780359	control	0
6	20CharsIsNotEnough	16	10	25	779	907	972	9	193	0.8755188	0.9307596	empathy	4

variable	explanation
AB	attacks before (pre-treatment)
AD	attacks during (the treatment period)
AA	attacks after (post-treatment)
CB	comments before
CD	comments during
CA	comments after
group	treatment group
IC	intervention count

```
  labs(title = "Intervention counts in treatment groups")+
  scale_x_continuous(breaks = seq(0,40,5))
```



Note there were much more empathetic interventions, this needs an explanation

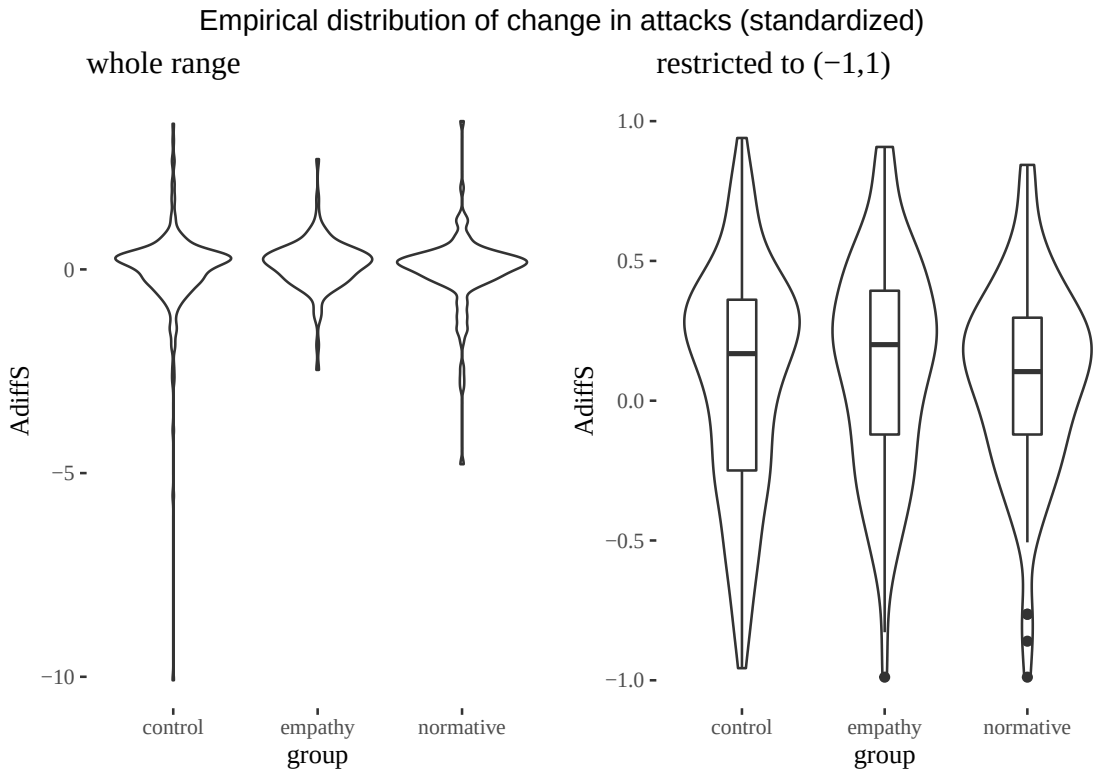
Second, when we look at the distribution of standardized difference in attacks, when restricted to (-1,1), the peaks of distributions are shifted a bit, with lowest median for the normative group, but not too much:

Question: intervention counts by group

```
violAdiffS <- ggplot(summaries, aes(x=group, y = AdiffS))+
  geom_violin() +theme_tufte()
violJoint <- ggarrange(violAdiffS+ggtitle("whole range"),
                violAdiffS + ylim(c(-1,1))+geom_boxplot(width = .2)+
                ggtitle("restricted to (-1,1)"))
```

```
## Warning: Removed 58 rows containing non-finite values (stat_ydensity).
## Warning: Removed 58 rows containing non-finite values (stat_boxplot).
```

```
violJointTitled <- annotate_figure(violJoint,
  top = text_grob("Empirical distribution of change in attacks (standardized)",
                  size = 12))
violJointTitled
```

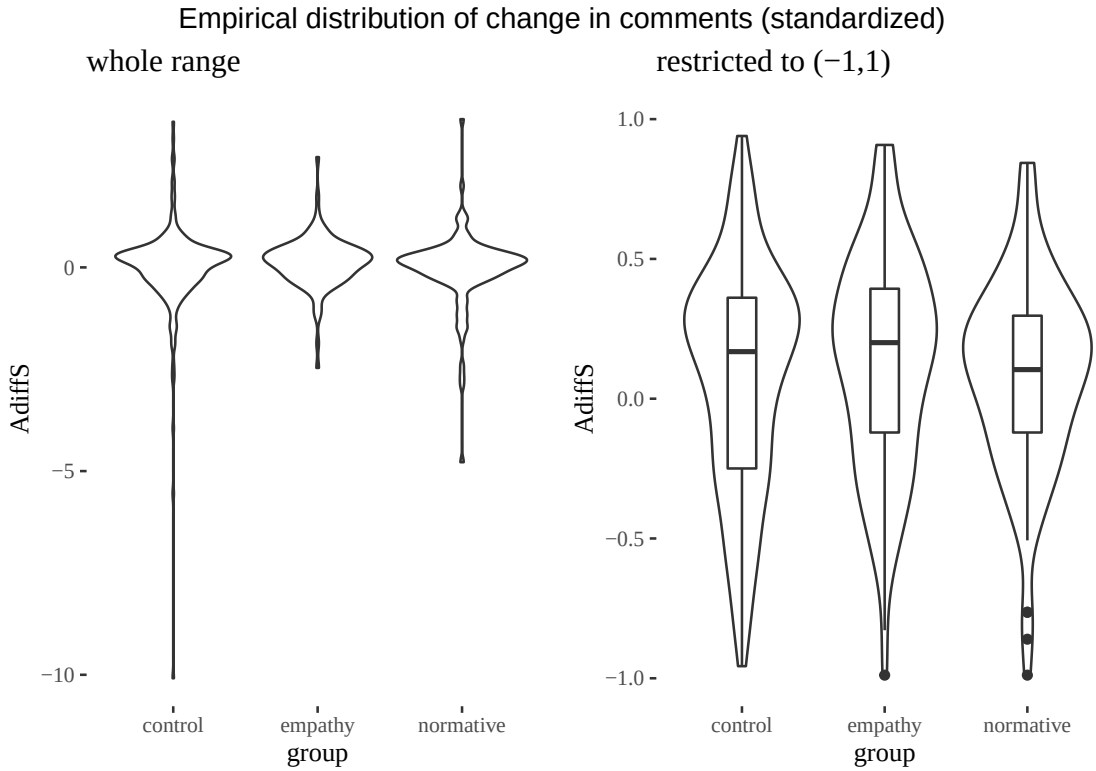


Analogous plot for comments does not reveal this slight downward shift for normative, but otherwise the visualisation migth suggest no strong impact of interventions on attacks, and no impact on comments.

```
violCdiffS <- ggplot(summaries, aes(x=group, y = CdiffS))+
  geom_violin() +theme_tufte()
violJointC <- ggarrange(violCdiffS+ggtitle("whole range"),
              violCdiffS + ylim(c(-1,1))+geom_boxplot(width = .2)+
              ggtitle("restricted to (-1,1)"))
```

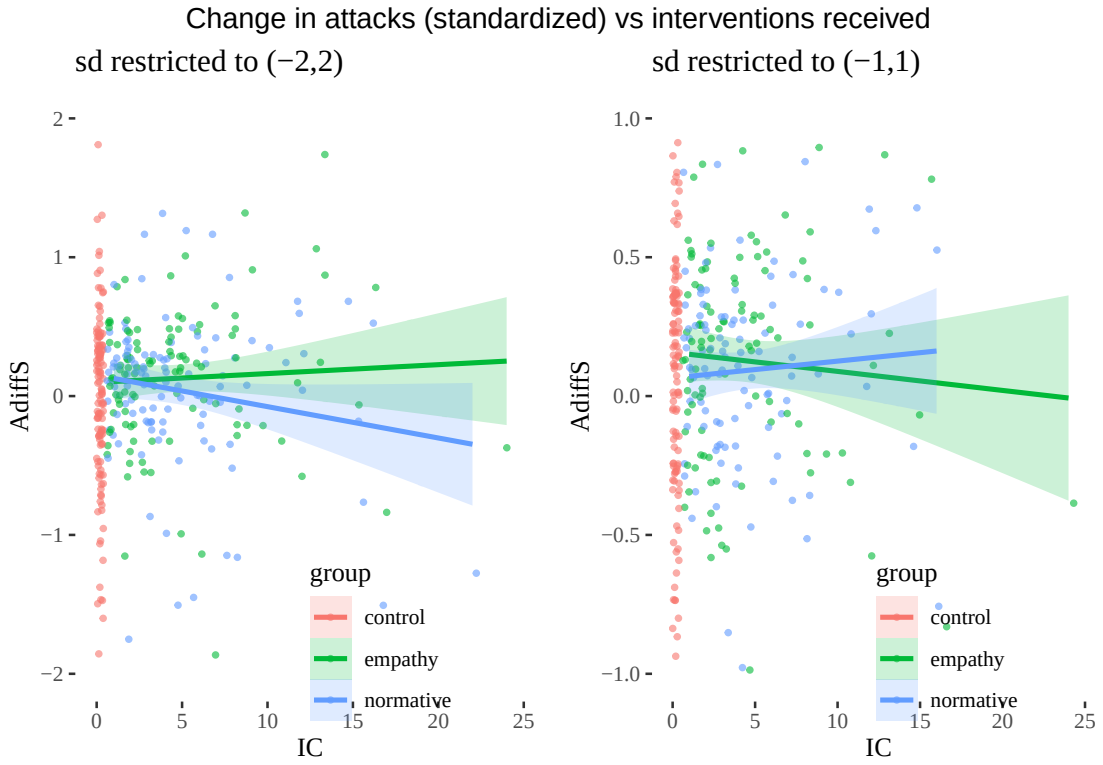
```
## Warning: Removed 90 rows containing non-finite values (stat_ydensity).
## Warning: Removed 90 rows containing non-finite values (stat_boxplot).
```

```
violJointCTitled <- annotate_figure(violJoint,
  top = text_grob("Empirical distribution of change in comments (standardized)",
                  size = 12))
violJointCTitled
```



However, plotting changes against intervention counts reveals that restricting attention to various activity levels drastically changes the regression lines.

```
icplot1 <- ggplot(summaries, aes(x = IC, y = AdiffS, color = group, fill = group))+
  geom_jitter(alpha = 0.6, size =.8)+theme_tufte()+
  geom_smooth(alpha = 0.2, method = "lm")+
  xlim(c(0,25))+ylim(c(-2,2))+
  ggtitle("sd restricted to (-2,2)")+
  theme(legend.position = c(0.65, 0.1))

icplot2 <-  ggplot(summaries, aes(x = IC, y = AdiffS, color = group, fill = group))+
  geom_jitter(alpha = 0.6, size =.8)+theme_tufte()+
  geom_smooth(alpha = 0.2, method = "lm")+
  xlim(c(0,25))+ylim(c(-1,1))+ggtitle("sd restricted to (-1,1)")+
  theme(legend.position = c(0.65, 0.1))

icplotJoint <- ggarrange(icplot1, icplot2)
icplotTitled <- annotate_figure(icplotJoint,
  top = text_grob("Change in attacks (standardized) vs interventions received",  size = 12))
icplotTitled
```



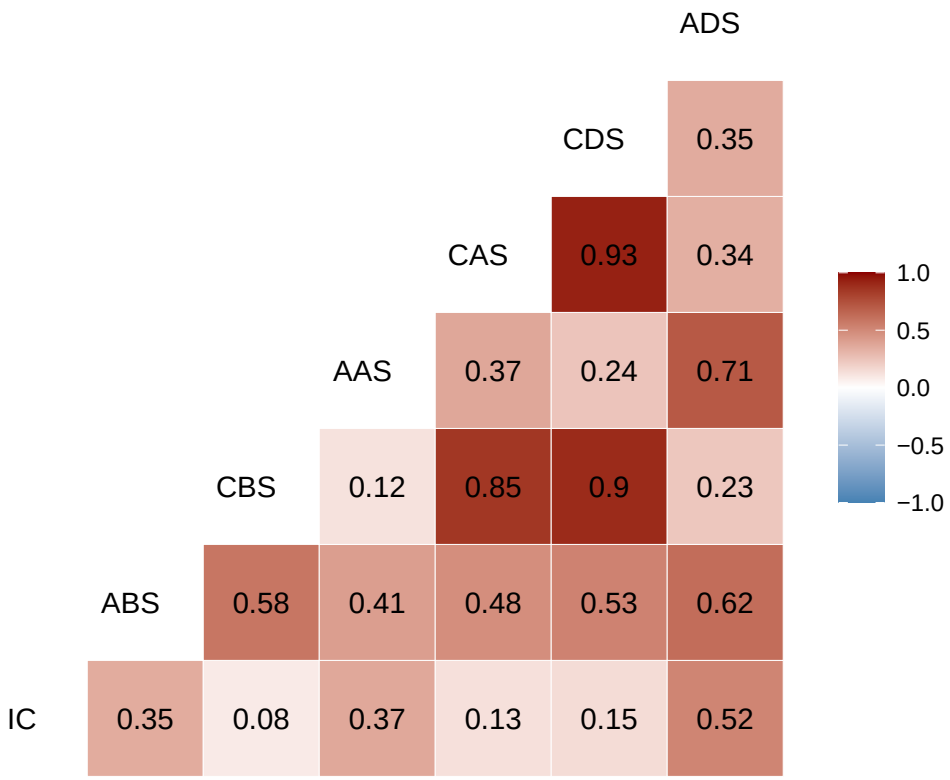
Some interactions are also suggested by the differences in linear smoothing when attention is restricted when it comes to change in comments.

```
icCplot1 <- ggplot(summaries, aes(x = IC, y = CdiffS, color = group, fill = group))+
  geom_jitter(alpha = 0.6, size =.8)+theme_tufte()+
  geom_smooth(alpha = 0.2, method = "lm")+
  xlim(c(0,25))+ylim(c(-2,2))+
  ggtitle("sd restricted to (-2,2)")+
  theme(legend.position = c(0.65, 0.1))

icCplot2 <-  ggplot(summaries, aes(x = IC, y = CdiffS, color = group, fill = group))+
  geom_jitter(alpha = 0.6, size =.8)+theme_tufte()+
  geom_smooth(alpha = 0.2, method = "lm")+
  xlim(c(0,25))+ylim(c(-1,1))+ggtitle("sd restricted to (-1,1)")+
  theme(legend.position = c(0.65, 0.1))

icCplotJoint <- ggarrange(icCplot1, icCplot2)
icCplotTitled <- annotate_figure(icCplotJoint,
  top = text_grob("Change in comments (standardized) vs interventions received",
   size = 12))
icCplotTitled
```

This suggests we should keep an eye out for interactions in the analysis, and that the intial comparison of means or medians between groups might be misleading if the effects in different volume groups are different and cancel each other.

Now, let's inspect correlations between the variables involved in the model:

```
summariesCorr <- select(summaries, IC, ABS, CBS, AAS, CAS, CDS, ADS)
ggcorr(summariesCorr, method = c("pairwise"),
       digits = 4, low = "steelblue", mid = "white",
       high = "darkred", midpoint =0,
       geom = "tile", label = TRUE, label_size=4, label_round =2, layout.exp =1,
       label_alpha = FALSE,hjust = 0.75)
```



This tells us that almost no predictors are strongly correlated, except for pairs CBS-CDS, so we drop CDS from the analysis and avoid using them in the same model to avoid multicolinearity issues. These are just comments during the intervention period, which, unsurprisingly are also a good proxy for comments before and comments after.

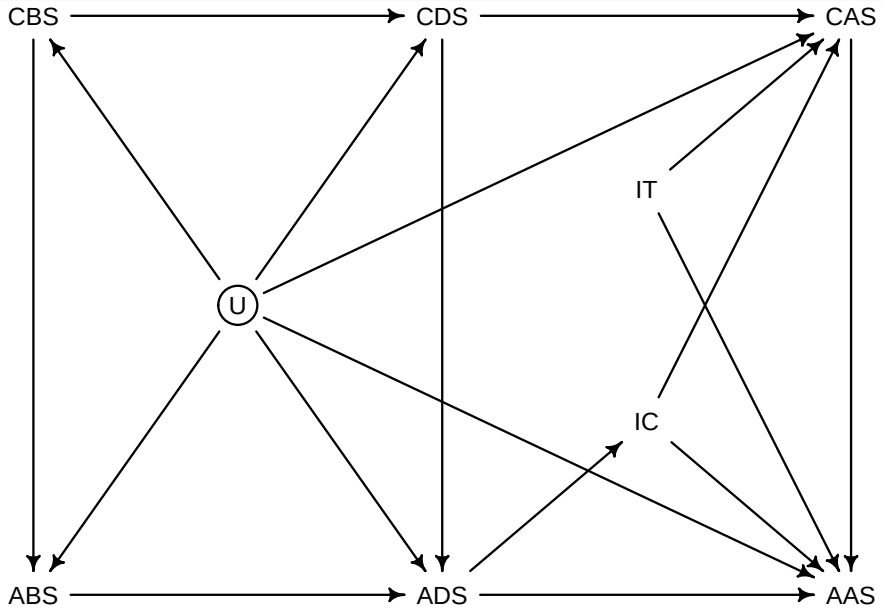
2 Causal inference

To identify the right variables to condition (or not condition) on to identify the causal effect of the interventions, we first need to think about the causal structure of the problem. Here's a plausible causal structure that we will be working with:

```
dag <- dagitty("
  dag{
  CDS -> ADS -> IC
                U [unobserved]
                U -> CBS -> ABS
                U -> ABS
                U -> CDS -> ADS
                U -> ADS
                U -> CAS -> AAS
                U -> AAS
                IC -> AAS
                IC -> CAS
                IT -> CAS
                IT -> AAS
                CBS -> CDS -> CAS
                ABS -> ADS -> AAS
                }")
coordinates(dag)
```

```
## $x
## AAS ABS ADS CAS CBS CDS  IC  IT   U
##  NA  NA  NA  NA  NA  NA  NA  NA  NA
##
## $y
## AAS ABS ADS CAS CBS CDS  IC  IT   U
##  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
coordinates( dag ) <- list( x=c(CBS=0,ABS=0,CDS=1,ADS=1, CAS = 2,
                    AAS = 2, IT = 1.5, IC = 1.5, U = .5) ,
y=c(CBS =0,ABS = 1,CDS = 0,ADS = 1, CAS = 0, AAS = 1,
    IT = .3, IC = .7, U =.5) )
drawdag(dag)
```



Comments during impact attacks during, which trigger interventions. Unmeasured user features cause comments before, which impact attacks before, and also attacks before directly. Comments during (their impact on ADS is aready included) impact attacks during during directly and comments after, which impact attacks after and attacks after directly. Intervention count impacts attacks after and comments after. The same directions of impact are included for intervention type. Finally, comments through time are connected causally, and so are attacks.

We already know not to condition on CDS if we condition on CAS or CBS. What else? IT has no bacwkard paths, but IC does. Let's identify all paths from IC to AAS:

```
paths(dag, from = c("IC"), to = "AAS")
```

```
## $paths
##  [1] "IC -> AAS"
##  [2] "IC -> CAS -> AAS"
##  [3] "IC -> CAS <- CDS -> ADS -> AAS"
##  [4] "IC -> CAS <- CDS -> ADS <- ABS <- CBS <- U -> AAS"
##  [5] "IC -> CAS <- CDS -> ADS <- ABS <- U -> AAS"
##  [6] "IC -> CAS <- CDS -> ADS <- U -> AAS"
##  [7] "IC -> CAS <- CDS <- CBS -> ABS -> ADS -> AAS"
##  [8] "IC -> CAS <- CDS <- CBS -> ABS -> ADS <- U -> AAS"
##  [9] "IC -> CAS <- CDS <- CBS -> ABS <- U -> AAS"
## [10] "IC -> CAS <- CDS <- CBS -> ABS <- U -> ADS -> AAS"
## [11] "IC -> CAS <- CDS <- CBS <- U -> AAS"
## [12] "IC -> CAS <- CDS <- CBS <- U -> ABS -> ADS -> AAS"
## [13] "IC -> CAS <- CDS <- CBS <- U -> ADS -> AAS"
## [14] "IC -> CAS <- CDS <- U -> AAS"
## [15] "IC -> CAS <- CDS <- U -> ABS -> ADS -> AAS"
## [16] "IC -> CAS <- CDS <- U -> ADS -> AAS"
## [17] "IC -> CAS <- CDS <- U -> CBS -> ABS -> ADS -> AAS"
## [18] "IC -> CAS <- IT -> AAS"
## [19] "IC -> CAS <- U -> AAS"
## [20] "IC -> CAS <- U -> ABS -> ADS -> AAS"
## [21] "IC -> CAS <- U -> ABS <- CBS -> CDS -> ADS -> AAS"
## [22] "IC -> CAS <- U -> ADS -> AAS"
## [23] "IC -> CAS <- U -> CBS -> ABS -> ADS -> AAS"
## [24] "IC -> CAS <- U -> CBS -> CDS -> ADS -> AAS"
## [25] "IC -> CAS <- U -> CDS -> ADS -> AAS"
## [26] "IC -> CAS <- U -> CDS <- CBS -> ABS -> ADS -> AAS"
## [27] "IC <- ADS -> AAS"
## [28] "IC <- ADS <- ABS <- CBS -> CDS -> CAS -> AAS"
## [29] "IC <- ADS <- ABS <- CBS -> CDS -> CAS <- IT -> AAS"
```

```
## [30] "IC <- ADS <- ABS <- CBS -> CDS -> CAS <- U -> AAS"
## [31] "IC <- ADS <- ABS <- CBS -> CDS <- U -> AAS"
## [32] "IC <- ADS <- ABS <- CBS -> CDS <- U -> CAS -> AAS"
## [33] "IC <- ADS <- ABS <- CBS -> CDS <- U -> CAS <- IT -> AAS"
## [34] "IC <- ADS <- ABS <- CBS <- U -> AAS"
## [35] "IC <- ADS <- ABS <- CBS <- U -> CAS -> AAS"
## [36] "IC <- ADS <- ABS <- CBS <- U -> CAS <- IT -> AAS"
## [37] "IC <- ADS <- ABS <- CBS <- U -> CDS -> CAS -> AAS"
## [38] "IC <- ADS <- ABS <- CBS <- U -> CDS -> CAS <- IT -> AAS"
## [39] "IC <- ADS <- ABS <- U -> AAS"
## [40] "IC <- ADS <- ABS <- U -> CAS -> AAS"
## [41] "IC <- ADS <- ABS <- U -> CAS <- IT -> AAS"
## [42] "IC <- ADS <- ABS <- U -> CBS -> CDS -> CAS -> AAS"
## [43] "IC <- ADS <- ABS <- U -> CBS -> CDS -> CAS <- IT -> AAS"
## [44] "IC <- ADS <- ABS <- U -> CDS -> CAS -> AAS"
## [45] "IC <- ADS <- ABS <- U -> CDS -> CAS <- IT -> AAS"
## [46] "IC <- ADS <- CDS -> CAS -> AAS"
## [47] "IC <- ADS <- CDS -> CAS <- IT -> AAS"
## [48] "IC <- ADS <- CDS -> CAS <- U -> AAS"
## [49] "IC <- ADS <- CDS <- CBS -> ABS <- U -> AAS"
## [50] "IC <- ADS <- CDS <- CBS -> ABS <- U -> CAS -> AAS"
## [51] "IC <- ADS <- CDS <- CBS -> ABS <- U -> CAS <- IT -> AAS"
## [52] "IC <- ADS <- CDS <- CBS <- U -> AAS"
## [53] "IC <- ADS <- CDS <- CBS <- U -> CAS -> AAS"
## [54] "IC <- ADS <- CDS <- CBS <- U -> CAS <- IT -> AAS"
## [55] "IC <- ADS <- CDS <- U -> AAS"
## [56] "IC <- ADS <- CDS <- U -> CAS -> AAS"
## [57] "IC <- ADS <- CDS <- U -> CAS <- IT -> AAS"
## [58] "IC <- ADS <- U -> AAS"
## [59] "IC <- ADS <- U -> ABS <- CBS -> CDS -> CAS -> AAS"
## [60] "IC <- ADS <- U -> ABS <- CBS -> CDS -> CAS <- IT -> AAS"
## [61] "IC <- ADS <- U -> CAS -> AAS"
## [62] "IC <- ADS <- U -> CAS <- IT -> AAS"
## [63] "IC <- ADS <- U -> CBS -> CDS -> CAS -> AAS"
## [64] "IC <- ADS <- U -> CBS -> CDS -> CAS <- IT -> AAS"
## [65] "IC <- ADS <- U -> CDS -> CAS -> AAS"
## [66] "IC <- ADS <- U -> CDS -> CAS <- IT -> AAS"
##
## $open
##  [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## [37]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## [49] FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## [61]  TRUE FALSE  TRUE FALSE  TRUE FALSE
```

Crucially, all backdoor paths go through ADS, which then becomes either a fork or a pipe, so all backdoor paths can be closed by conditioning on ADS. Moreover there is only one directed indirect path, it goes through CAS, so we should not condition on it if we are to identify causal effect on attacks mediated by impact on comments (unless we care about the direct effect of IC and IT on AAS, but that's a separate question). This is in line with the adjustment set identified algorithmically.

```
adjustmentSets(dag, exposure = c("IC", "IT"), outcome = "AAS", type = "all")
```

```
## { ADS }
## { ABS, ADS }
## { ADS, CBS }
## { ABS, ADS, CBS }
## { ADS, CDS }
## { ABS, ADS, CDS }
## { ADS, CBS, CDS }
## { ABS, ADS, CBS, CDS }
## { ADS, U }
## { ABS, ADS, U }
## { ADS, CBS, U }
## { ABS, ADS, CBS, U }
## { ADS, CDS, U }
## { ABS, ADS, CDS, U }
## { ADS, CBS, CDS, U }
## { ABS, ADS, CBS, CDS, U }
```

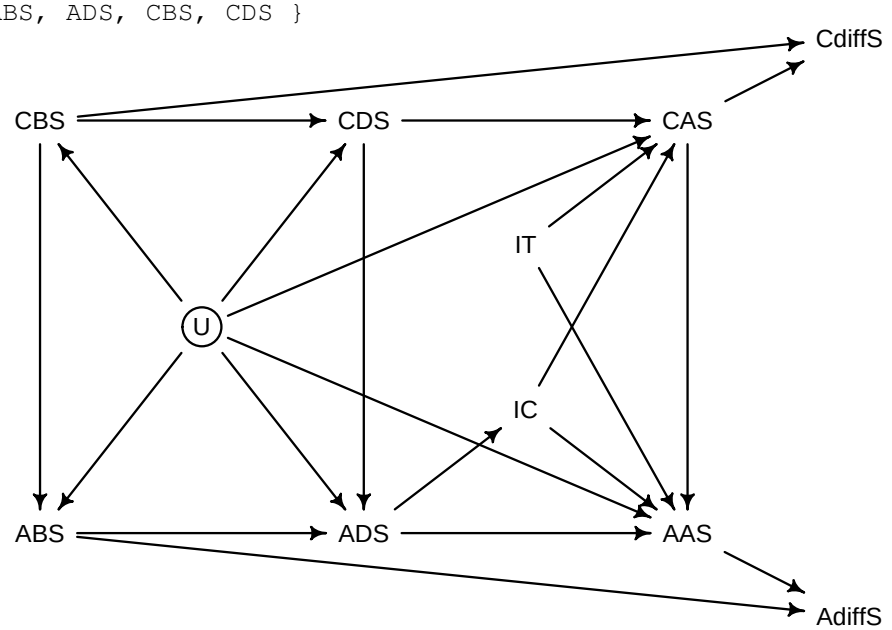
The situation is different when it comes to evaluating the *direct* effect of intervention. Then, we also need to block indirect causal paths from the intervention to the outcome. For such an evaluation we

need to also condition on CAS, which is what we will do when we turn to the study of the direct effects of the interventions.

In fact, we will be predicting the difference between attacks before and after, and the difference between comments, before and after. Let's add them to the dag to double-check our selection of variables.

```
dag2 <- dagitty("
  dag{
                CDS -> ADS -> IC
                U [unobserved]
                U -> CBS -> ABS
                U -> ABS
                U -> CDS -> ADS
                U -> ADS
                U -> CAS -> AAS
                U -> AAS
                IC -> AAS
                IC -> CAS
                IT -> CAS
                IT -> AAS
                CBS -> CDS -> CAS
                ABS -> ADS -> AAS
                ABS -> AdiffS
                AAS -> AdiffS
                CBS -> CdiffS
                CAS -> CdiffS
                }")
coordinates( dag2 ) <- list( x=c(CBS=0,ABS=0,CDS=1,ADS=1,
    CAS = 2, AAS = 2, IT = 1.5, IC = 1.5, U = .5, AdiffS= 2.5, CdiffS = 2.5) ,
y=c(CBS =0,ABS = 1,CDS = 0,ADS = 1, CAS = 0, AAS = 1, IT = .3, IC = .7, U =.5, Cdif

drawdag(dag2)
adjustmentSets(dag2, exposure = c("IC", "IT"), outcome = "AdiffS", type = "canonica
```

```
## { ABS, ADS, CBS, CDS }
```



Finding a maximal sensible set (canonical) of covariates suggests inlcuding CDS and ABS. As already discussed, we do not include CDS because of its strong correlation with CBS. We also do not condition on ABS—not only because it has a pretty strong correlation with another predictor (ADS), but rather mainly because it is used to define the output variable. In such a set-up, of course that a model including

ABS would have better predictive power, but since a definitional connection is present, thinking that its inclusion in the model tells us something about causality would be misled.

It's open season for the other variables (and interactions between them), and our decision to include them in the model will be guided by information-theoretic criteria of predictive power.

We will focus on a class of additive models where the outcome variable is normally distributed around the predicted mean, which is a linear function of predictors (possibly with some interactions). To spoil the story, we will end up using a model, whose specification is as follows:

$$
\begin{aligned}
\text{AdiffS} &\sim \text{Norm}(\mu, \sigma) \\
\mu_i &= \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}_i} + \beta_{\text{IC}}[\text{group}_i] \times \text{IC} + \\
&\quad + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} + \beta_{\text{CBS}}[\text{group}_i] \times \text{CBS} \\
\alpha &\sim \text{Norm}(0, .3) \\
\beta_{\text{ADS}}[\text{group}_i] &\sim \text{Norm}(0, .3) \\
\beta_{\text{group}_i} &\sim \text{Norm}(0, .3) \\
\beta_{\text{IC}}[\text{group}_i] &\sim \text{Norm}(0, .3) \\
\beta_{\text{ADSIC}} &\sim \text{Norm}(0, .3) \\
\beta_{\text{CBS}}[\text{group}_i] &\sim \text{Norm}(0, .3)
\end{aligned}
$$

That is, we take the resulting mean to be the result of the general average (α) and the impact of the following coefficients: group-specific coefficient for ADS, group coefficient, group-specific coefficient for IC, interaction coefficient for ADS and IC, and group-specific coeffient for CBS. This is plausible prima facie which group a user belongs to might have impact on how attacks during the treatment is related to attacks after, the role of the intervention count, and the role of comments before. Moreover, the levels of agressive behavior displayed by the user during treament might have impact on the role played by the intervention count. Later on we will see that there are information-theoretic reasons to include these interactions.

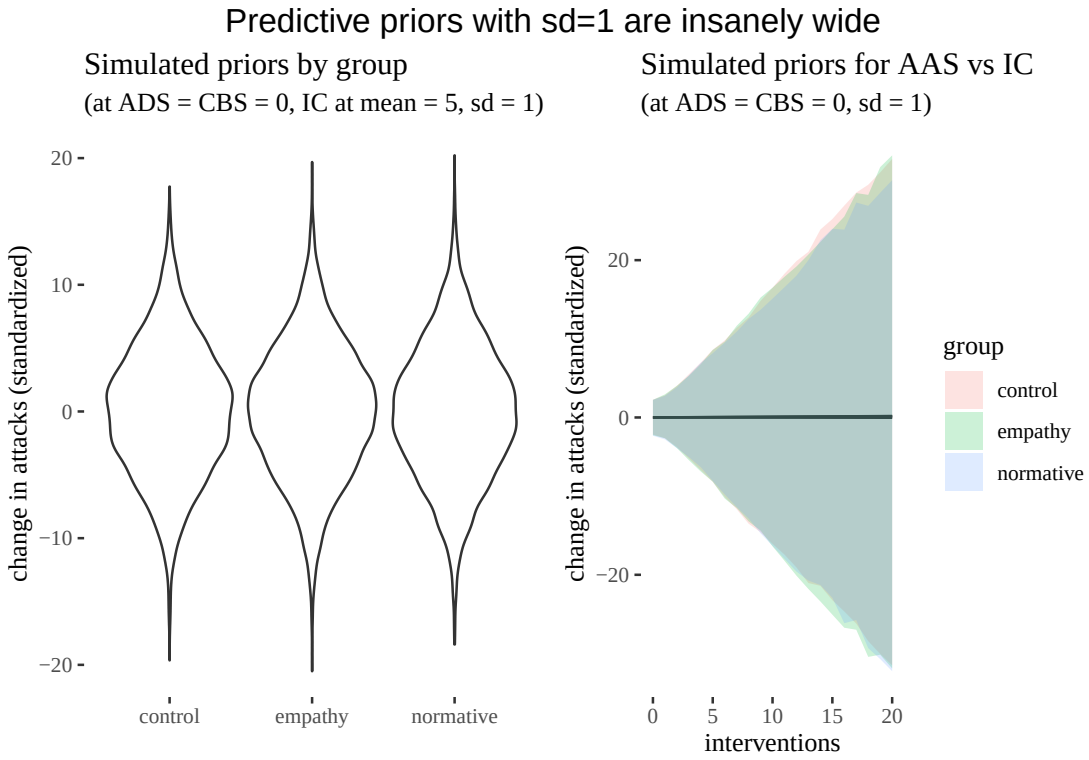
Now for the priors. One might be suspicious of $\sigma = .3$ we employed and suggest using standard normal distributions with $\sigma = 1$ instead. However, a quick prior predictive check shows that this results in insanely wide priors that are competely unrealistic. (For computational reasons, instead of running the simulations, we load pre-compiled models, but we include the code used to build them).

```
# building model with sd=1
# InteractionsModelDiffSD1 <- ulam(
#   alist(
#     AdiffS ~ dnorm( mu, sigma ),
#     mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC+
#     bADSIC * ADS * IC+ bCBS[groupID] *CBS,
#     a ~ dnorm (0,1),
#     bADS[groupID] ~ dnorm(0,1),
#     bADSIC ~ dnorm(0,1),
#     bCBS[groupID] ~ dnorm(0,1),
#     bIT[groupID] ~ dnorm(0,1),
#     bIC[groupID] ~ dnorm(0,1),
#     sigma  ~ dexp(1)
#   ),
#   data = summaries
# )
#
# saveRDS(InteractionsModelDiffSD1, file = "models/InteractionsModelDiffSD1.rds")
InteractionsModelDiffSD1 <- readRDS(file = "models/InteractionsModelDiffSD1.rds")


#now model with prior sd = .3
# InteractionsModelDiff <- ulam(
#   alist(
#     AdiffS ~ dnorm( mu, sigma ),
#     mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
#     bADSIC * ADS * IC+ bCBS[groupID] *CBS,
#     a ~ dnorm (0,0.3),
```

```
#     bADS[groupID] ~ dnorm(0,.3),
#     bADSIC ~ dnorm(0,.3),
#     bCBS[groupID] ~ dnorm(0,.3),
#     bIT[groupID] ~ dnorm(0,.3),
#     bIC[groupID] ~ dnorm(0,.3),
#     sigma  ~ dexp(1)
#   ),
#   data = summaries
# )

#saveRDS(InteractionsModelDiff, file = "models/InteractionsModelDiff.rds")

InteractionsModelDiff <- readRDS(file = "models/InteractionsModelDiff.rds")

##prior predictive checks sd =1
ADS <- 0
CBS <- 0
groupID <- 1:3
IC <- 5  #mean for interventions in treatment
data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, IC =  IC)
prior <- extract.prior(InteractionsModelDiffSD1, n = 1e4)
mu <- link( InteractionsModelDiffSD1 , post=prior , data=data )
colnames(mu) <- levels(summaries$group)
muLong <- melt(mu)
colnames(muLong) <- c("id", "group", "AdiffS")

priorGroupsSD1 <- ggplot(muLong)+
  geom_violin(aes(x = group, y = AdiffS))+
  theme_tufte()+xlab("")+
  labs(title = "Simulated priors by group",
  subtitle = "(at ADS = CBS = 0, IC at mean = 5, sd = 1)")+
  ylab("change in attacks (standardized)")

ADS <- 0
CBS <- 0
groupID <- 1:3
IC <- 0:20
data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, IC =  IC)

prior <- extract.prior(InteractionsModelDiffSD1, n = 1e4)
```

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

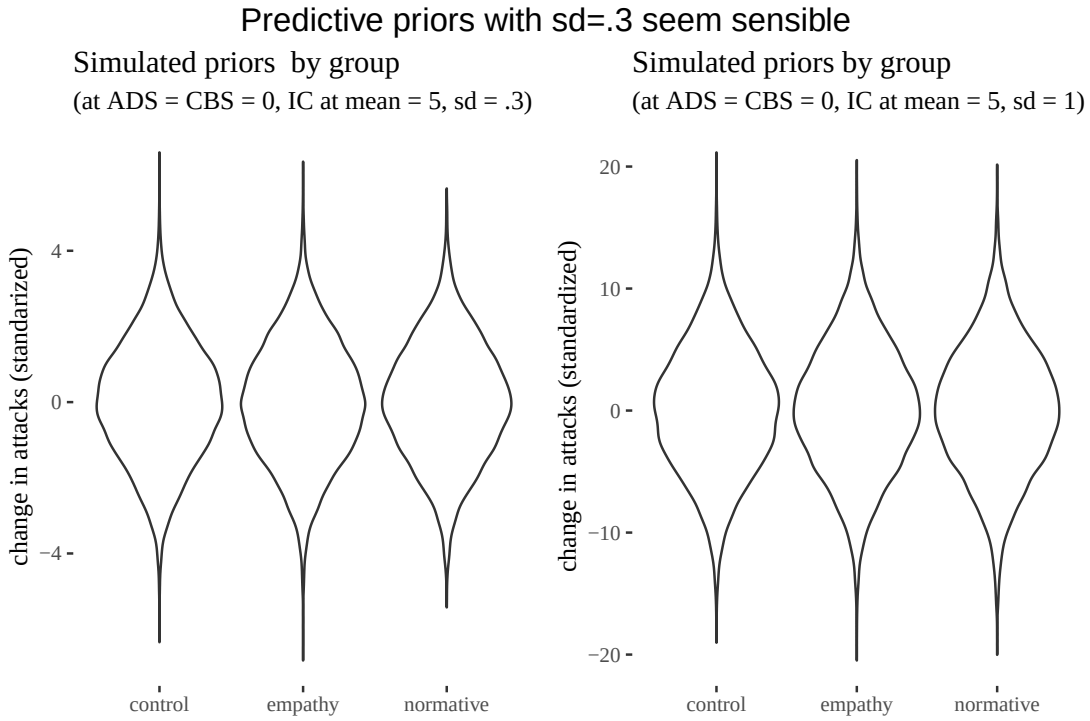
```
mu <- link(InteractionsModelDiffSD1 , post=prior , data=data )
mu.mean <- apply( mu , 2, mean )
mu.HPDI <- data.frame(t(apply( mu , 2 , HPDI )))
priorDF <- cbind(data, mu.mean, mu.HPDI)
priorDF$groupID <- as.factor(groupID)
levels(priorDF$groupID) <- c("control", "empathy", "normative")
colnames(priorDF)[2]<- "group"


priorICSD1  <- ggplot(priorDF, aes(x = IC, y  = mu.mean,  fill = group))+
  geom_line()+geom_ribbon(aes(ymin = X.0.89, ymax = X0.89.), alpha = 0.2)+
  theme_tufte()+ylab("change in attacks (standardized)")+
  labs(title = "Simulated priors for AAS vs IC",
      subtitle = "(at ADS = CBS = 0, sd = 1)")+xlab("interventions")


priorJoint1 <- ggarrange(priorGroupsSD1,priorICSD1, ncol = 2)
priorJoint1Titled <- annotate_figure(priorJoint1,
  top = text_grob("Predictive priors with sd=1 are insanely wide",
                  size = 14))
priorJoint1Titled
```



Some experimentation leads to the value of $\sigma = .3$, which leads to the following priors:

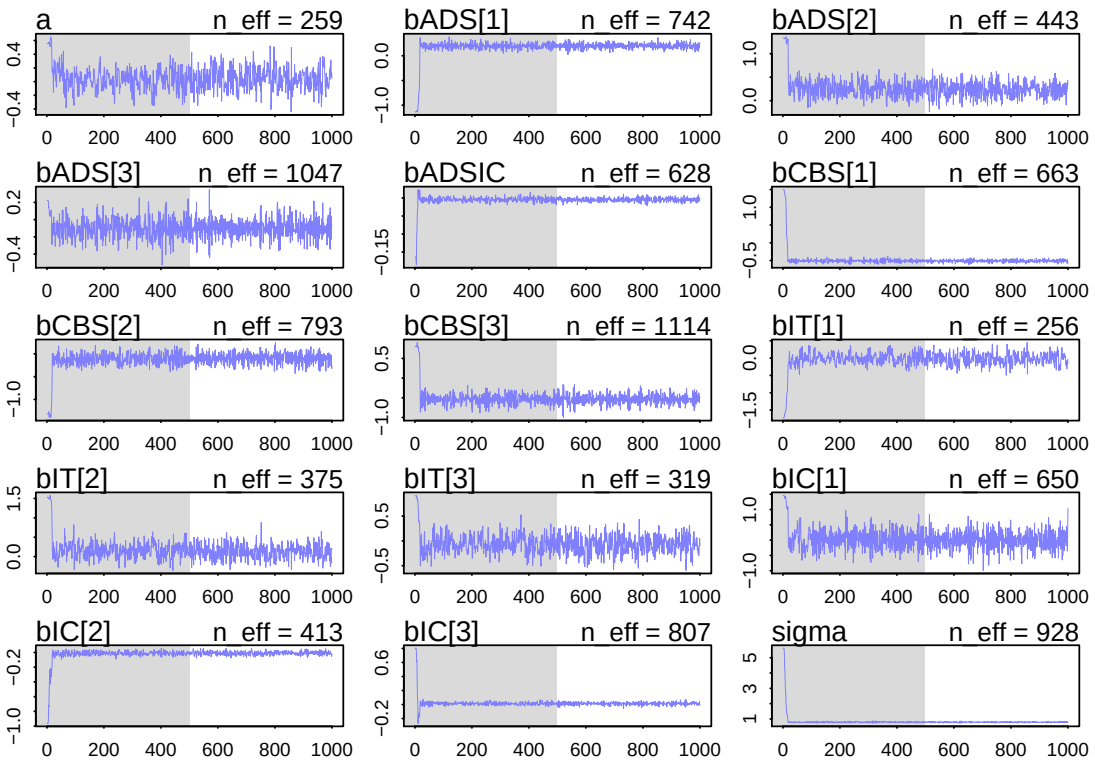
```
#prior predictive check sd =.3
ADS <- 0
CBS <- 0
groupID <- 1:3
IC <- 5  #mean for interventions in treatment
data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, IC =  IC)
prior <- extract.prior(InteractionsModelDiff, n = 1e4)
mu <- link(InteractionsModelDiff , post=prior , data=data )
colnames(mu) <- levels(summaries$group)
muLong <- melt(mu)
colnames(muLong) <- c("id", "group", "AdiffS")
head(muLong)

priorGroupSD03 <- ggplot(muLong)+
  geom_violin(aes(x = group, y = AdiffS))+theme_tufte()+
  xlab("")+
  labs(title = "Simulated priors  by group",
  subtitle = "(at ADS = CBS = 0, IC at mean = 5, sd = .3)")+
  ylab("change in attacks (standarized)")

ADS <- 0
CBS <- 0
groupID <- 1:3
IC <- 5  #mean for interventions in treatment
data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, IC =  IC)
prior <- extract.prior(InteractionsModelDiffSD1, n = 1e4)
mu <- link( InteractionsModelDiffSD1 , post=prior , data=data )
colnames(mu) <- levels(summaries$group)
muLong <- melt(mu)
colnames(muLong) <- c("id", "group", "AdiffS")
head(muLong)

priorICSD03 <- ggplot(muLong)+
  geom_violin(aes(x = group, y = AdiffS))+
  theme_tufte()+xlab("")+
  labs(title = "Simulated priors by group",
  subtitle = "(at ADS = CBS = 0, IC at mean = 5, sd = 1)")+
  ylab("change in attacks (standardized)")

priorJoint03 <- ggarrange(priorGroupSD03,priorICSD03, ncol = 2)
```

```
priorJoint03Titled <- annotate_figure(priorJoint03,
  top = text_grob("Predictive priors with sd=.3 seem sensible",
                  size = 14))
priorJoint03Titled
```



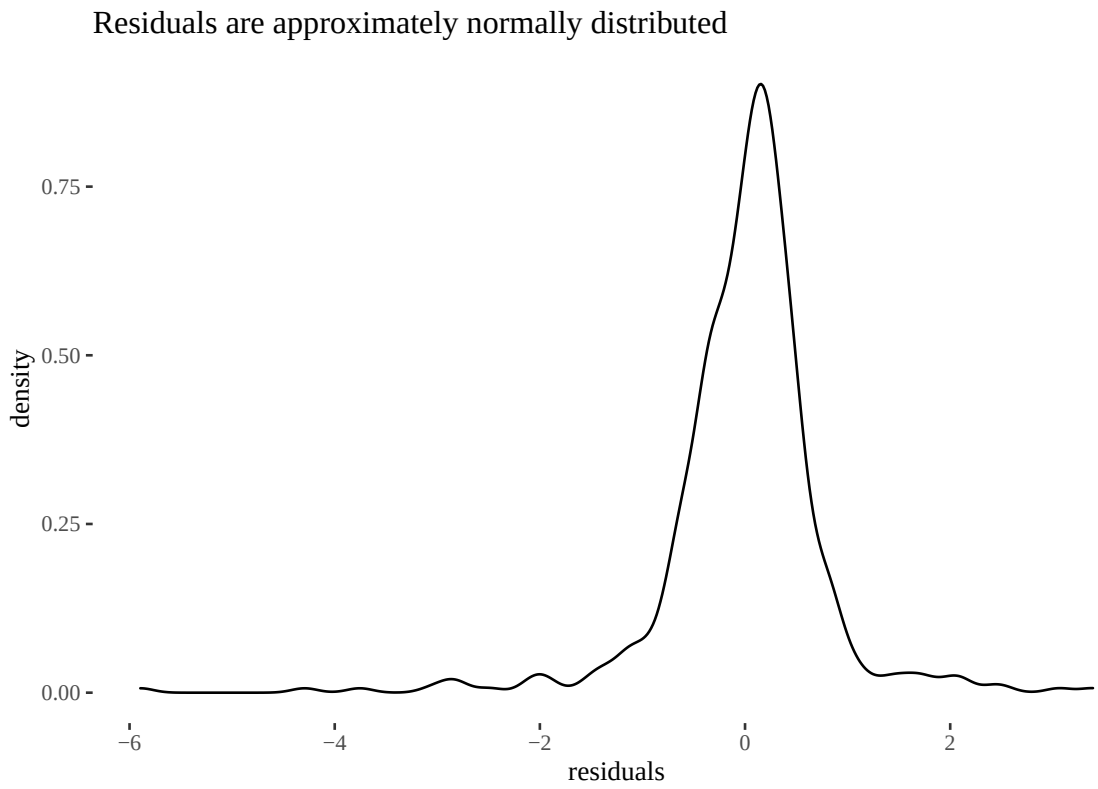
Now, some model diagnostics before we move on. What we are witnessing is (1) stationarity (the chains stay mostly in the most probable regions), (2) good mixing (they explore a range of options in the beginning), and (3) convergence (they stabilize as they progress).

```
traceplot( InteractionsModelDiff )
```



Finally, let's inspect the distribution of residuals. That is, we calculate all predictions, their distance from the actual values, and inspect the distribution of the distances:

```
mu <- link(InteractionsModelDiff)
mu_mean <- apply( mu , 2 , mean )
mu_resid <- summaries$AdiffS - mu_mean
ggplot()+geom_density(aes(x = mu_resid))+theme_tufte()+
  ggtitle("Residuals are approximately normally distributed")+xlab("residuals")
```



3 Model selection

How did we get to this fairly complicated model though? Once preliminary causal considerations guided our restrictions on variable selection, we proceed by building models of increasing complexity, and comparing them in terms of Widely Acceptable Information Criterion. The models differ mostly in the underlying linear formulae. For computational ease we will here use quadratic approximations, while in the final analysis we will deploy Hamiltonian Monte Carlo. The names are meant to decode the model structure: the predictors are listed before dashes, whereas interactions are listed after dashes.

$$\mu_i = \alpha \quad \text{(Null)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}} \times \text{ADS} \quad \text{(ADS)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}} \times \text{ADS} + \beta_{\text{IC}} \times \text{IC} \quad \text{(ADSIC)}$$

$$\mu_i = \beta_{\text{group}[i]} \quad \text{(IT)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}} \times \text{ADS} + \beta_{\text{group}[i]} \quad \text{(ADSIT)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}} \times \text{ADS} + \beta_{\text{group}[i]} + \beta_{\text{IC}} \times \text{IC} \quad \text{(ADSITIC)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}} \times \text{ADS} + \beta_{\text{group}[i]} + \beta_{\text{IC}} \times \text{IC} + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} \quad \text{(ADSITIC-ADSIC)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}[i]} + + \beta_{\text{IC}} \times \text{IC} + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} \quad \text{(ADSITIC-ADSIC-ADSIT)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}[i]} \quad \text{(ADSIT-ADSIT)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}[i]} + \beta_{\text{IC}}[\text{group}_i] \times \text{IC} + + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} \quad \text{(ADSITIC-ADSIT-ITIC-ADSIC)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}[i]} + \beta_{\text{IC}}[\text{group}_i] \times \text{IC} + + \beta_{\text{CBS}} \times \text{CBS} + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} \quad \text{(ADSITICCBS-ITIC-ADSIC)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}_i} + \beta_{\text{IC}}[\text{group}_i] \times \text{IC} + + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} + \beta_{\text{CBS}}[\text{group}_i] \times \text{CBS} \quad \text{(Final)}$$

$$\mu_i = \alpha + \beta_{\text{ADS}}[\text{group}_i] \times \text{ADS} + \beta_{\text{group}_i} + \beta_{\text{IC}}[\text{group}_i] \times \text{IC} + + \beta_{\text{ADSIC}} \times \text{ADS} \times \text{IC} + \beta_{\text{CBS}}[\text{group}_i] \times \text{CBS} + \beta_{\text{CBSIC}} \times \text{CBS} \times \text{IC} \quad \text{(tooFAr)}$$

$$\quad (1)$$

```
null <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu ~ dnorm (0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ADS <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  a + bADS * ADS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ADSIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  a + bADS * ADS+ bIC * IC,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,0.3),
    bIC ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


IT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  bIT[groupID] ,
    bIT[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
```

```
  ),
  data = summaries
)


ADSIT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID],
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID] + bIC * IC,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID] + bIC * IC + bADSIC * ADS * IC,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSIC_ADSIT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC * IC + bADSIC * ADS * IC,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSIT_ADSIT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] ,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    #bADSIC ~ dnorm(0,.5),
    bIT[groupID] ~ dnorm(0,.3),
    #bIC ~ dnorm(0,.5),
```

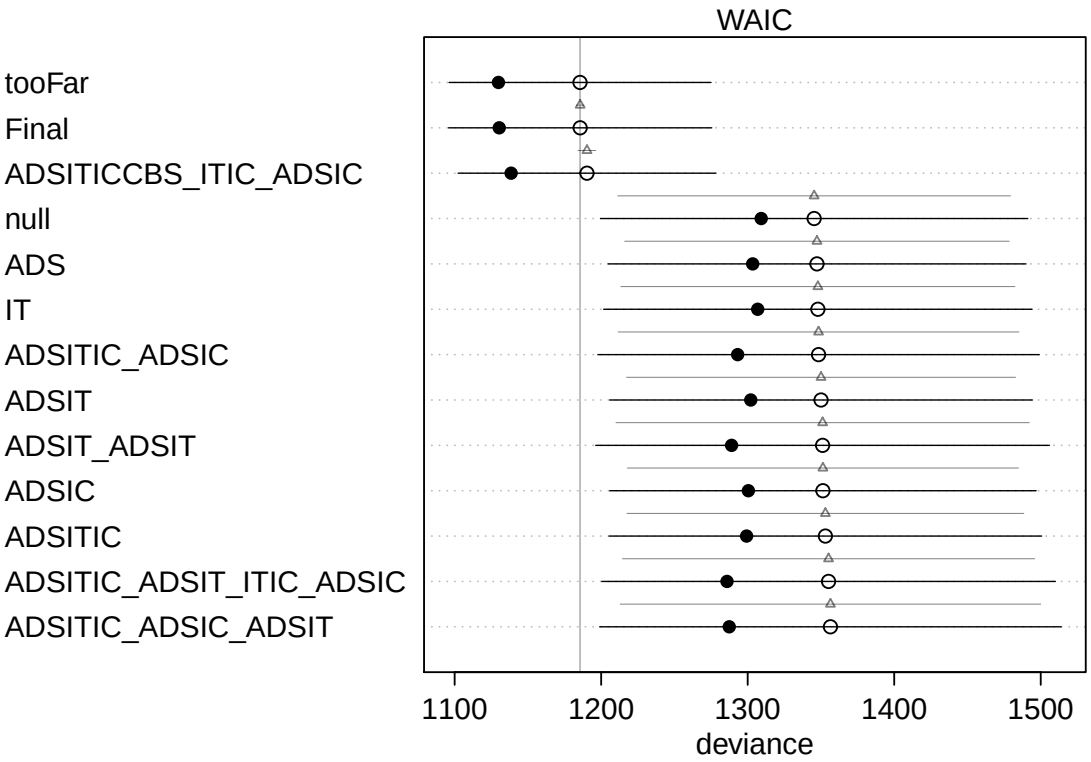
```
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSIT_ITIC_ADSIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITICCBS_ITIC_ADSIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS *CBS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCBS ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


Final <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)



tooFar <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
     bCBSIC ~ dnorm(0, .3),
    sigma  ~ dexp(1)
  ),
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
tooFar	1185.575	89.393	0.000	NA	27.846	0.487
Final	1185.657	89.827	0.082	2.594	27.614	0.467
ADSITICCBS_ITIC_ADSIC	1190.304	87.920	4.729	6.055	25.882	0.046
null	1345.358	145.805	159.783	134.039	18.061	0.000
ADS	1347.281	142.655	161.706	131.261	21.935	0.000
IT	1347.916	146.200	162.341	134.462	20.545	0.000
ADSITIC_ADSIC	1348.366	150.638	162.791	136.811	27.604	0.000
ADSIT	1350.067	144.264	164.492	132.764	23.985	0.000
ADSIT_ADSIT	1351.139	154.826	165.564	141.033	31.062	0.000
ADSIC	1351.343	145.539	165.768	133.454	25.443	0.000
ADSITIC	1353.020	147.645	167.445	135.441	26.884	0.000
ADSITIC_ADSIT_ITIC_ADSIC	1355.187	154.826	169.613	140.659	34.652	0.000
ADSITIC_ADSIC_ADSIT	1356.479	157.522	170.905	143.538	34.533	0.000

```
  data = summaries
)
```

```
comparison<- compare(null,ADS,ADSIC,IT,ADSIT,ADSITIC,ADSITIC_ADSIC,
                     ADSITIC_ADSIC_ADSIT,ADSIT_ADSIT,ADSITIC_ADSIT_ITIC_ADSIC,
                     ADSITICCBS_ITIC_ADSIC,Final, tooFar)
mykable(round(data.frame(comparison ),3))
```

```
plot(comparison)
```



The three models that stand out differ in including CBS as a predictor. Moreover the final model includes an interaction between treatment group and CBS. Adding a further interaction between CBS and IC takes us too far. *WAIC*-based weighing assigns the weight of 83% to the final model, and the standard errors for the difference in WAIC for the top three models is fairly low, so we will employ the top model (Final) in further analyses.

	mean	sd	X5.5.	X94.5.	n_eff	Rhat4
a	0.0414089	0.1562753	-0.2037740	0.2796946	228.1593	0.9980513
bADS[1]	0.1979211	0.0568494	0.1086423	0.2907669	768.4216	1.0013137
bADS[2]	0.2484756	0.1546028	0.0036955	0.4942282	526.1494	0.9995921
bADS[3]	-0.1097667	0.1014546	-0.2678764	0.0628934	418.0876	0.9980791
bADSIC	-0.0042060	0.0051643	-0.0122338	0.0040408	361.1195	0.9983620
bCBS[1]	-0.5142899	0.0414267	-0.5776731	-0.4463925	722.3116	0.9991926
bCBS[2]	-0.0919148	0.1245614	-0.2965873	0.1059034	671.6673	0.9998084
bCBS[3]	-0.5302840	0.1088835	-0.7023272	-0.3593766	823.7157	0.9980583
bIT[1]	-0.0212337	0.1630179	-0.2603815	0.2295494	219.0190	0.9980137
bIT[2]	0.1482970	0.1857528	-0.1246834	0.4367149	304.7859	0.9980345
bIT[3]	-0.0686005	0.1776783	-0.3448157	0.2099344	295.3412	0.9980482
bIC[1]	-0.0037630	0.3161707	-0.5084313	0.4981378	829.2326	1.0011451
bIC[2]	-0.0117801	0.0256771	-0.0529960	0.0290304	488.1809	0.9994090
bIC[3]	0.0118155	0.0185521	-0.0192521	0.0412303	612.3810	0.9986804
sigma	0.7945882	0.0277774	0.7535960	0.8434971	693.5845	0.9982674

4 Inspecting the model and effect sizes

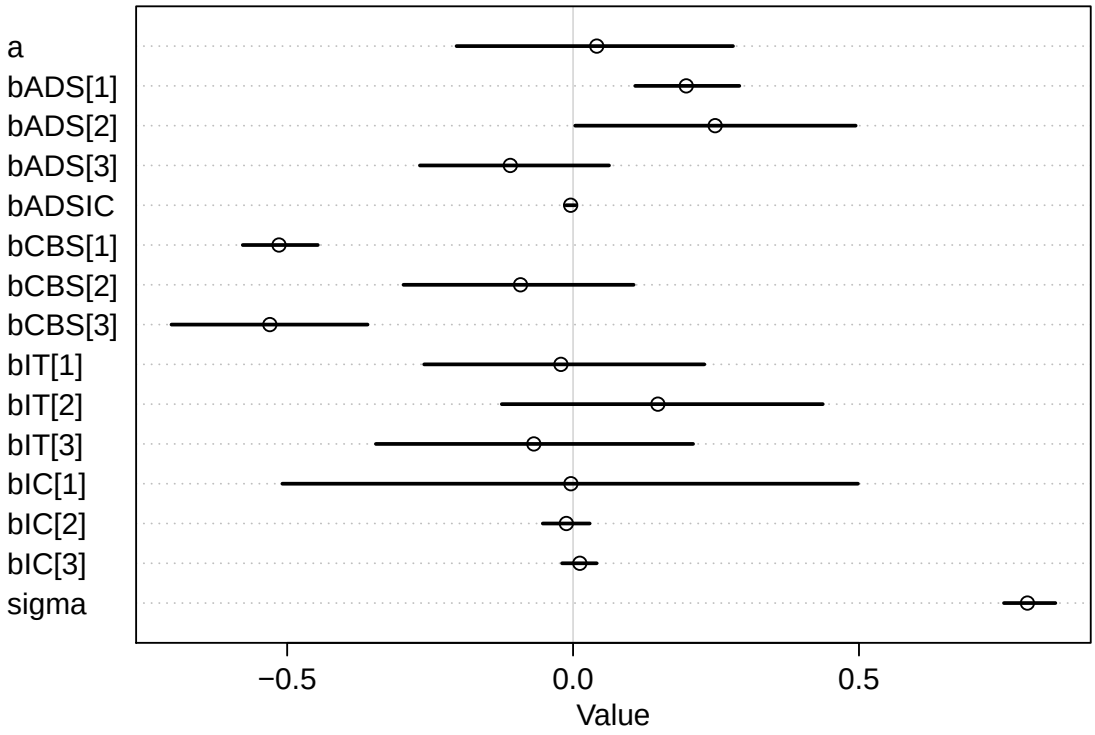
We start by using the Final model formula to build a model, this time using Hamiltonian Monte Carlo. We leave the code commented out and load a pre-compiled model for computational convenience

```
# FinalHMC <- ulam(
#   alist(
#     AdiffS ~ dnorm( mu, sigma ),
#     mu <- a + bADS[groupID] * ADS +  bIT[groupID] +
#     bIC[groupID] * IC + bADSIC * ADS * IC+
#     bCBS[groupID] *CBS,
#     a ~ dnorm (0,0.3),
#     bADS[groupID] ~ dnorm(0,.3),
#     bADSIC ~ dnorm(0,.3),
#     bCBS[groupID] ~ dnorm(0,.3),
#     bIT[groupID] ~ dnorm(0,.3),
#     bIC[groupID] ~ dnorm(0,.3),
#     sigma  ~ dexp(1)
#   ),
#   data = summaries
# )
# saveRDS(FinalHMC, file = "models/FinalHMC.rds")


#saveRDS(FinalHMC, file = "models/FinalHMC.rds")

FinalHMC <- readRDS(file = "models/FinalHMC.rds")
```

First, let's take a look at the best model coefficients.

```
mykable(data.frame(precis(FinalHMC, depth = 2)))
```

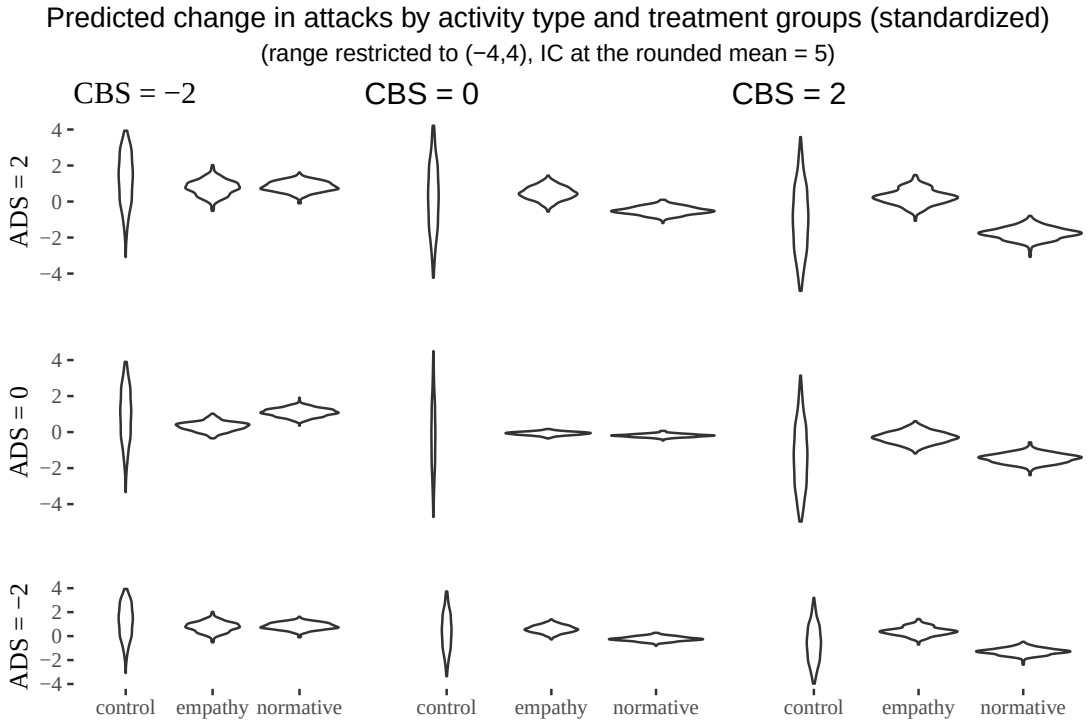
```
plot(precis(FinalHMC, depth = 2))
```



These, however, are notoriously hard to interpret in models with interactions. For this reason, it is better to plot predicted effects for various combinations of predictors.

```
visGroup <- function (model, ADS, CBS, xmin =2, ymax = -3)
{
groupID <- 1:3
IC <- 5
data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, IC =  IC)
posterior <- extract.samples(model, n = 1e5)
mu <- link( model, data=data )
colnames(mu) <- levels(summaries$group)
muLong <- melt(mu)
colnames(muLong) <- c("id", "group", "AdiffS")
means <-  round(apply(mu , 2 , mean ), 2)
mu_HPDI <- round(apply( mu , 2 , HPDI ),2)
means <- as.data.frame(means)
means$group <- rownames(means)
rownames(means) <- NULL
meansDisp <- cbind(means,t(as.data.frame(mu_HPDI)))
meansDisp <- meansDisp[,c(1,3,4)]

plot <- ggplot(muLong)+geom_violin(aes(x = group, y = AdiffS), alpha = 0.2)+
  xlab("")+
  labs(title = paste("ADS=", ADS, ", CBS=",  CBS,  sep = ""))+
  theme_tufte()+ylim(c(-4,4))
#+   annotation_custom(tableGrob(meansDisp), xmin=xmin,  ymax=ymax)
return(plot)
}


visGroupA2C_2 <- visGroup(model = FinalHMC, ADS = 2,CBS = -2)
visGroupA2C0 <- visGroup(model = FinalHMC, ADS = 2,CBS = 0 )
visGroupA2C2 <- visGroup(model = FinalHMC, ADS = 2,CBS = 2)

visGroupA0C_2 <- visGroup(model = FinalHMC, ADS = 0,CBS = -2 )
visGroupA0C0 <- visGroup(model = FinalHMC, ADS = 0,CBS = 0 )
visGroupA0C2 <-  visGroup(model = FinalHMC, ADS = 0,CBS = 2)

visGroupA2C_2 <-  visGroup(model = FinalHMC, ADS = 2,CBS = -2 )
visGroupA2C0 <- visGroup(model = FinalHMC, ADS = 2,CBS = 0 )
visGroupA2C2 <- visGroup(model = FinalHMC, ADS = 2,CBS = 2 )

visGroupJoint <- ggarrange(visGroupA2C_2+removeX + ggtitle("CBS = -2")+ylab("ADS = 2") , visGroupA2C0+th
```

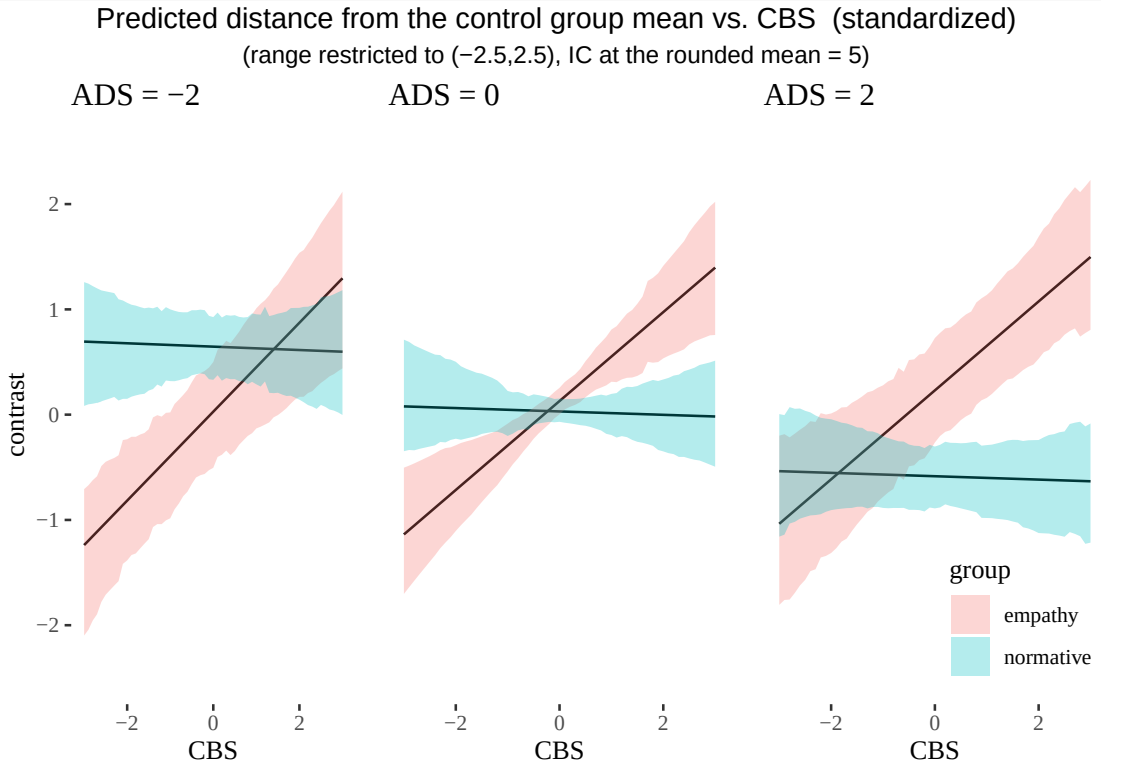
```
          visGroupA0C_2+removeX+ylab("ADS = 0")+ggtitle(""), visGroupA0C0+theme_void()+ggtitle(""), visG
          visGroupA2C_2+ylab("ADS = -2")+ggtitle(""), visGroupA2C0+removeY+ggtitle(""), visGroupA2C2+rem


visGroupJoint2 <- annotate_figure(visGroupJoint,
  top = text_grob("(range restricted to (-4,4), IC at the rounded mean = 5)",
                  size = 10))
visGroupJoint3 <- annotate_figure(visGroupJoint2,
  top = text_grob("Predicted change in attacks by activity type and treatment groups (standardized)",
                  size = 12))

visGroupJoint3
```



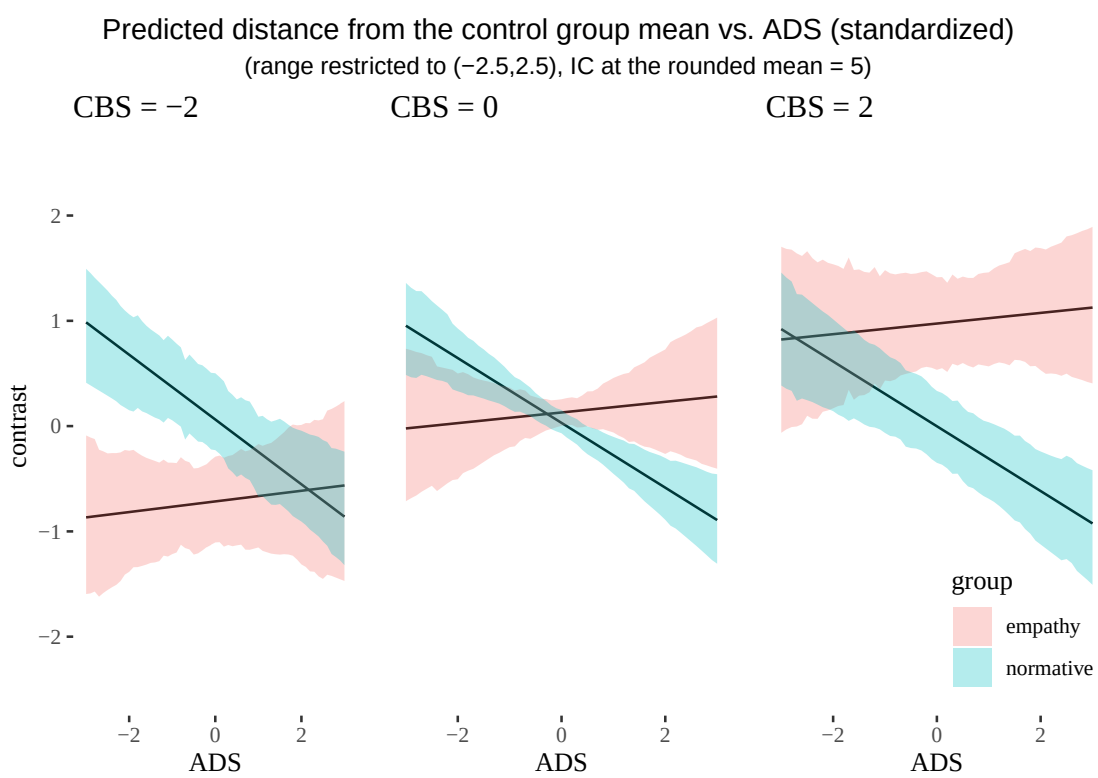
To gain more clarity, let's look at predicted contrasts, here understood as distances from the control group mean, by activity types, first versus CBS, then versus ADS.

```
visContrastsCBS <- function(model = FinalHMC, ADS = ADS , IC =  5,
                            CBS = seq(-3,3,by  = 0.1))
  {
  groupID <- 1:3
  data <- expand.grid(ADS, groupID, IC , CBS)
  colnames(data) <- c("ADS", "groupID", "IC", "CBS")
  posterior <- extract.samples(model, n = 1e5)
  link( model, data=data )
  mu <- link( model, data=data )
  means <-  round(apply(mu , 2 , mean ), 4)
  HPDIs <- round(apply( mu , 2 , HPDI ),4)
  visContrast <- cbind(data,means,t(as.data.frame(HPDIs)))

  ones <- 3 * (1:(nrow(visContrast)/3))-2
  twos <- 3 * (1:(nrow(visContrast)/3))-1
  threes <- 3 * (1:(nrow(visContrast)/3))

  colnames(visContrast)[c(6,7)] <- c("low", "high")
  contrast <- numeric(nrow(visContrast))
  cLow <- numeric(nrow(visContrast))
  cHigh <- numeric(nrow(visContrast))
  for(i in threes){
  contrast[i] <- visContrast$means[i] - visContrast$means[i-2]
  }
  for(i in twos){
  contrast[i] <- visContrast$means[i] - visContrast$means[i-1]
```

```
  }
  visContrast$contrast <- contrast
  visContrast$shift <-  visContrast$contrast - visContrast$means
  for(i in ones){
  visContrast$shift[i] <- 0
  }
  visContrast$cLow <- visContrast$low + visContrast$shift
  visContrast$cHigh <- visContrast$high + visContrast$shift

  visContrast$group = rep(c("control", "empathy", "normative"),
                          nrow(visContrast)/3)

  visContrastTreatment <- visContrast[groupID !=1,]

  return(ggplot(visContrastTreatment, aes(x = CBS, y = contrast, fill = group ))+
           geom_line(se = FALSE)+
            geom_ribbon(mapping =
         aes(ymin = cLow, ymax = cHigh),
          alpha = .3)+
           theme_tufte())
}


visContrastCBSJoint <- ggarrange(visContrastsCBS(FinalHMC,ADS = -2)+
      ggtitle("ADS = -2")+ylim(c(-2.5,2.5))+ scale_fill_discrete(guide=FALSE),
          visContrastsCBS(FinalHMC,ADS = 0)+ggtitle("ADS = 0")+
        ylim(c(-2.5,2.5))+ scale_fill_discrete(guide=FALSE)+
        removeY,
        visContrastsCBS(FinalHMC,ADS = 2)+ggtitle("ADS = 2")+
        ylim(c(-2.5,2.5))+ theme(legend.position = c(0.75, 0.15))+
        removeY, ncol = 3)

visContrastCBSJoint2 <- annotate_figure(visContrastCBSJoint,
      top = text_grob("(range restricted to (-2.5,2.5), IC at the rounded mean = 5)",
                                  size = 10))
visContrastCBSJoint3 <- annotate_figure(visContrastCBSJoint2,
top = text_grob("Predicted distance from the control group mean vs. CBS  (standardized)",
                                  size = 12))

visContrastCBSJoint3
```



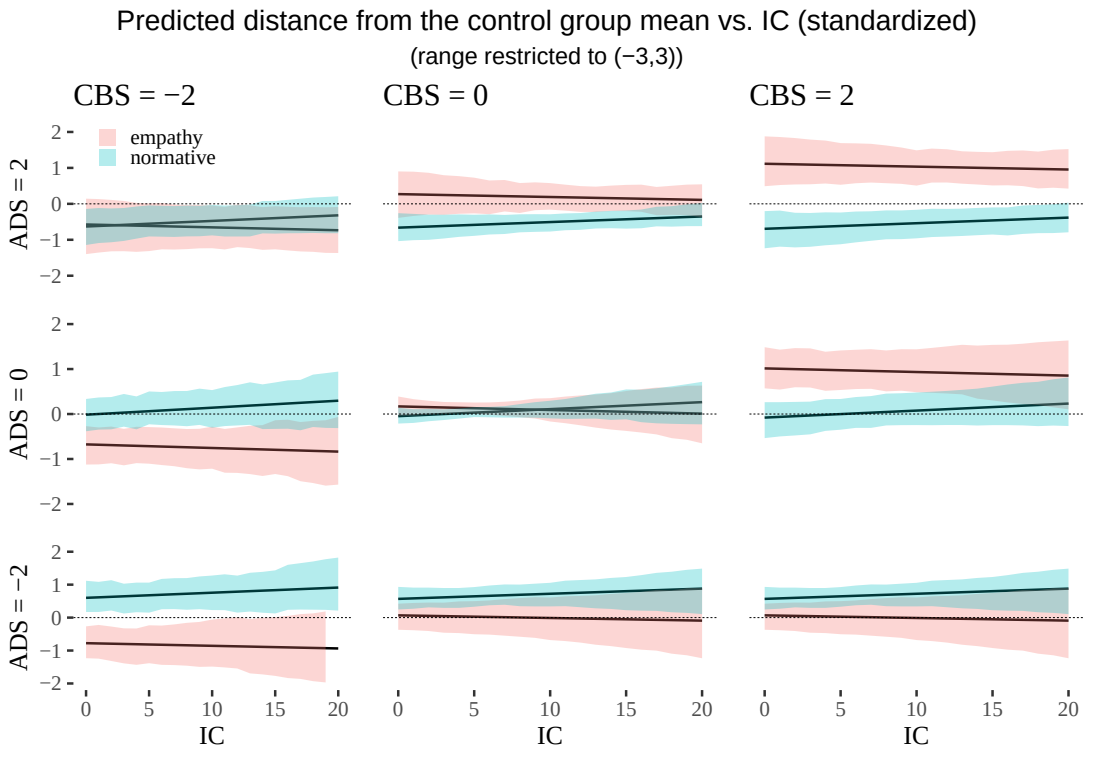
```
visContrastsADS <- function(model = FinalHMC, CBS = CBS , IC =  5,
                            ADS = seq(-3,3,by  = 0.1))
{
  data <- expand.grid(CBS, groupID, IC , ADS)
  colnames(data) <- c("CBS", "groupID", "IC", "ADS")
  posterior <- extract.samples(model, n = 1e5)
  mu <- link( model, data=data )
  means <-  round(apply(mu , 2 , mean ), 4)
  HPDIs <- round(apply( mu , 2 , HPDI ),4)
  visContrastADS <- cbind(data,means,t(as.data.frame(HPDIs)))


  ones <- 3 * (1:(nrow(visContrastADS)/3))-2
  twos <- 3 * (1:(nrow(visContrastADS)/3))-1
  threes <- 3 * (1:(nrow(visContrastADS)/3))

  colnames(visContrastADS)[c(6,7)] <- c("low", "high")
  contrastADS <- numeric(nrow(visContrastADS))
  for(i in threes){
    contrastADS[i] <- visContrastADS$means[i] - visContrastADS$means[i-2]
  }
  for(i in twos){
    contrastADS[i] <- visContrastADS$means[i] - visContrastADS$means[i-1]
  }
  visContrastADS$contrast <- contrastADS
  visContrastADS$shift <-  visContrastADS$contrast - visContrastADS$means
  for(i in ones){
    visContrastADS$shift[i] <- 0
  }
  visContrastADS$cLow <- visContrastADS$low + visContrastADS$shift
  visContrastADS$cHigh <- visContrastADS$high + visContrastADS$shift

  visContrastADS$group = rep(c("control", "empathy", "normative"),
                             nrow(visContrastADS)/3)
  visContrastTreatmentADS <- visContrastADS[groupID !=1,]

  return(ggplot(visContrastTreatmentADS, aes(x = ADS, y = contrast, fill = group ))+
           geom_line(se = FALSE) +
  geom_ribbon(mapping = aes(ymin = cLow, ymax = cHigh),
       alpha = .3) +theme_tufte())
}



visContrastADSJoint <- ggarrange(visContrastsADS(FinalHMC,CBS = -2)+ggtitle("CBS = -2")
                                 +ylim(c(-2.5,2.5))+ scale_fill_discrete(guide=FALSE),
                                 visContrastsADS(FinalHMC,CBS = 0)+ggtitle("CBS = 0")
                                 +ylim(c(-2.5,2.5))+ scale_fill_discrete(guide=FALSE)+
                                   removeY,
                                 visContrastsADS(FinalHMC,CBS = 2)+ggtitle("CBS = 2")
                                 +ylim(c(-2.5,2.5))+
                                   theme(legend.position = c(0.75, 0.15))+
                                   removeY, ncol = 3)

visContrastADSJoint2 <- annotate_figure(visContrastADSJoint,
top = text_grob("(range restricted to (-2.5,2.5), IC at the rounded mean = 5)",
                                                    size = 10))
visContrastADSJoint3 <- annotate_figure(visContrastADSJoint2,
top = text_grob("Predicted distance from the control group mean vs. ADS (standardized)",
                                                    size = 12))

visContrastADSJoint3
```



Now, let's inspect the impact of intervention counts by treatment type by looking at contrasts (distances from the control group mean) with 89% HPDIs by IC. Notice the predicted effect of IC is weaker than group membership, so for visibility the y-axis has a smaller range. Also, not enough data was available to reliably estimate uncertainty for IC above 20, hence the restriction on the x-axis (arleady at lower values, lack of estimates is visible for the more extreme covariate settings).

```
visContrastsIC <- function(model = FinalHMC, CBS = CBS ,
                           IC =  seq(0,20,by = 1), ADS = ADS)
{
  groupID <- 1:3
  data <- expand.grid(CBS, groupID, IC , ADS)
  data
  colnames(data) <- c("CBS", "groupID", "IC", "ADS")
  posterior <- extract.samples(model, n = 1e5)
  mu <- link( model, data=data )
  means <-  round(apply(mu , 2 , mean ), 4)
  HPDIs <- round(apply( mu , 2 , HPDI ),4)
  visContrastIC <- cbind(data,means,t(as.data.frame(HPDIs)))

  ones <- 3 * (1:(nrow(visContrastIC)/3))-2
  twos <- 3 * (1:(nrow(visContrastIC)/3))-1
  threes <- 3 * (1:(nrow(visContrastIC)/3))

  colnames(visContrastIC)[c(6,7)] <- c("low", "high")
  contrastIC <- numeric(nrow(visContrastIC))
  for(i in threes){
    contrastIC[i] <- visContrastIC$means[i] - visContrastIC$means[i-2]
  }
  for(i in twos){
    contrastIC[i] <- visContrastIC$means[i] - visContrastIC$means[i-1]
  }
  visContrastIC$contrast <- contrastIC
  visContrastIC$shift <-  visContrastIC$contrast - visContrastIC$means
  for(i in ones){
    visContrastIC$shift[i] <- 0
  }
  visContrastIC$cLow <- visContrastIC$low + visContrastIC$shift
  visContrastIC$cHigh <- visContrastIC$high + visContrastIC$shift

  visContrastIC$group = rep(c("control", "empathy", "normative"),
                            nrow(visContrastIC)/3)
```

```
  visContrastTreatmentIC <- visContrastIC[groupID !=1,]

  return(ggplot(visContrastTreatmentIC, aes(x = IC, y = contrast, fill = group ))+
           geom_line(se=FALSE)+
    geom_ribbon(mapping = aes(ymin = cLow, ymax = cHigh), alpha = .3)+
    ylim(c(-2,2)) +theme_tufte()+geom_hline(yintercept = 0, lty =2, size = 0.1))
}


visContrastsICJoint <- ggarrange(
visContrastsIC(ADS = 2, CBS = -2)+removeX+
               theme(legend.position = c(0.3, 0.9),
                     legend.key.size = unit(.3, 'cm'),
                     legend.key.height = unit(.3, 'cm'),
                     legend.key.width = unit(.3, 'cm'),
                     legend.title= element_blank())+
  ggtitle("CBS = -2")+
  ylab("ADS = 2"),
    visContrastsIC(ADS = 2, CBS = 0)+removeY+removeX+ scale_fill_discrete(guide=FALSE)+
  ggtitle("CBS = 0"),
    visContrastsIC(ADS = 2, CBS = 2)+removeY+removeX+ggtitle("CBS = 2")+ scale_fill_discrete(guide=FALSE
visContrastsIC(ADS = 0, CBS = -2)+removeX+ scale_fill_discrete(guide=FALSE)+
  ylab("ADS = 0"),
    visContrastsIC(ADS = 0, CBS = 0)+removeY+removeX+
  scale_fill_discrete(guide=FALSE),
    visContrastsIC(ADS = 0, CBS = 2)+removeY+removeX+ scale_fill_discrete(guide=FALSE),
visContrastsIC(ADS = -2, CBS = -2)+ scale_fill_discrete(guide=FALSE)+
  ylab("ADS = -2"),
    visContrastsIC(ADS = -2, CBS = 0)+removeY+ scale_fill_discrete(guide=FALSE),
    visContrastsIC(ADS = -2, CBS = 0)+removeY+ scale_fill_discrete(guide=FALSE),
ncol = 3, nrow = 3
)

visContrastsICJoint2 <- annotate_figure(visContrastsICJoint,
top = text_grob("(range restricted to (-3,3))",
                size = 10))
visContrastsICJoint3 <- annotate_figure(visContrastsICJoint2,
 top = text_grob("Predicted distance from the control group mean vs. IC (standardized)",
                size = 12))

visContrastsICJoint3
```



5 Direct effect

Models for the evaluation of direct effect need to close the indirect causal path from the treatment variables to the output, and they do so by conditioning on CAS. Again, we face model selection. We repeat all the model structures from the previous section, except for now, each model is extended with CAS as a predictor. This time the model structure we called tooFar turns out to do better. We also consider extending it with interactions between CAS and IT and IC (jointly and separately), but this results in no further improvement.

```
nullDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bCAS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ADSDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  a + bADS * ADS+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,0.3),
    bCAS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ADSICDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  a + bADS * ADS+ bIC * IC+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,0.3),
    bIC ~ dnorm(0,0.3),
    bCAS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ITDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <-  bIT[groupID] + bCAS * CAS,
    bIT[groupID] ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)

ADSITDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID]+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bIT[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)
```

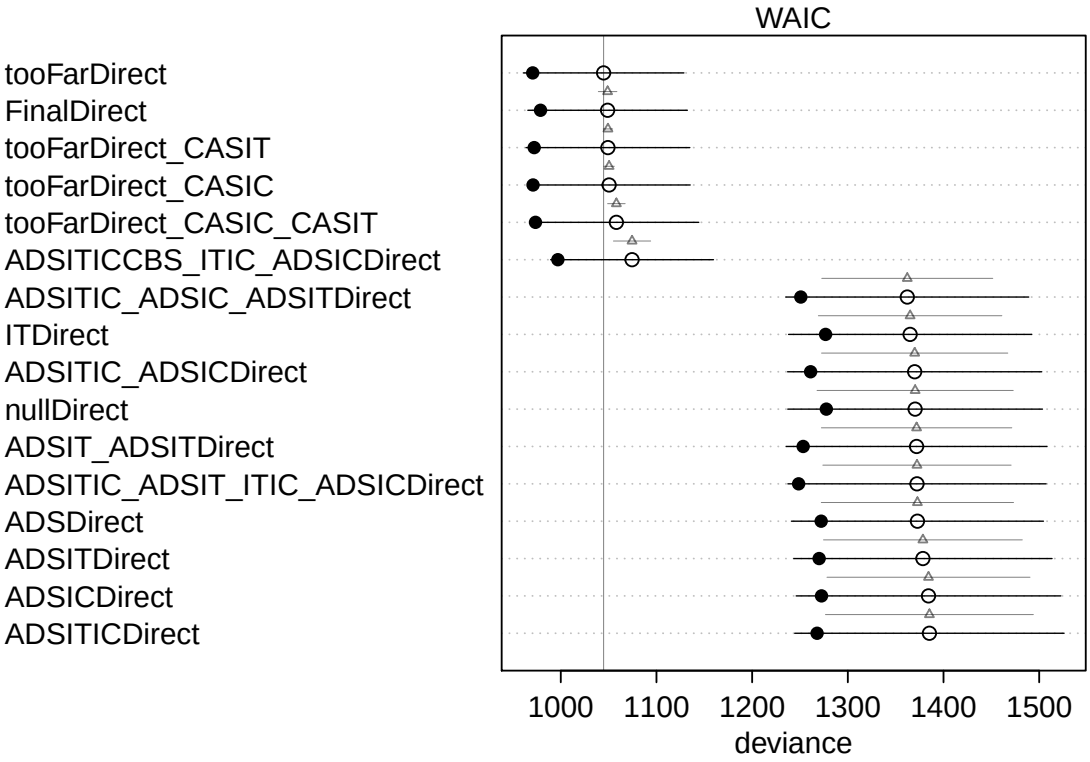
```
ADSITICDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID] + bIC * IC+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSICDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS * ADS +  bIT[groupID] + bIC * IC + bADSIC * ADS * IC+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bADSIC ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSIC_ADSITDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC * IC + bADSIC * ADS * IC+ bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSIT_ADSITDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    #bADSIC ~ dnorm(0,.5),
    bIT[groupID] ~ dnorm(0,.3),
    #bIC ~ dnorm(0,.5),
    bCAS ~ dnorm(0,0.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITIC_ADSIT_ITIC_ADSICDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
```

```
    bCAS ~ dnorm(0,0.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


ADSITICCBS_ITIC_ADSICDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS *CBS + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCBS ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


FinalDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)



tooFarDirect <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC + bCAS * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    bCBSIC ~ dnorm(0, .3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)



tooFarDirect_CASIT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
```

```
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC + bCAS[groupID] * CAS,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS[groupID] ~ dnorm(0,0.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    bCBSIC ~ dnorm(0, .3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)



tooFarDirect_CASIC <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC + bCAS * CAS + bCASIC * CAS * IC,
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS ~ dnorm(0,0.3),
    bCASIC ~ dnorm(0,0.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    bCBSIC ~ dnorm(0, .3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)


tooFarDirect_CASIC_CASIT <- quap(
  alist(
    AdiffS ~ dnorm( mu, sigma ),
    mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
      bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC + bCAS[groupID] * CAS + bCASIC * CAS * I
    a ~ dnorm (0,0.3),
    bADS[groupID] ~ dnorm(0,.3),
    bADSIC ~ dnorm(0,.3),
    bCAS[groupID] ~ dnorm(0,0.3),
    bCASIC ~ dnorm(0,0.3),
    bCBS[groupID] ~ dnorm(0,.3),
    bIT[groupID] ~ dnorm(0,.3),
    bIC[groupID] ~ dnorm(0,.3),
    bCBSIC ~ dnorm(0, .3),
    sigma  ~ dexp(1)
  ),
  data = summaries
)







comparisonDirect<- compare(nullDirect,ADSDirect,ADSICDirect,ITDirect,ADSITDirect,ADSITICDirect,ADSITIC_A
                     ADSITICCBS_ITIC_ADSICDirect,FinalDirect, tooFarDirect,tooFarDirect_CASIT,
                     tooFarDirect_CASIC,tooFarDirect_CASIC_CASIT)


mykable(round(data.frame(comparisonDirect),3))%>% kable_styling(latex_options = c("striped","scale_down"
```

```
plot(comparisonDirect)
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
tooFarDirect	1044.746	83.935	0.000	NA	37.053	0.777
FinalDirect	1048.925	83.369	4.178	9.720	35.057	0.096
tooFarDirect_CASIT	1049.258	85.612	4.512	4.952	38.533	0.081
tooFarDirect_CASIC	1050.488	84.774	5.741	5.460	39.743	0.044
tooFarDirect_CASIC_CASIT	1058.089	86.136	13.343	9.164	42.266	0.001
ADSITICCBS_ITIC_ADSICDirect	1074.474	85.119	29.727	19.463	38.768	0.000
ADSITIC_ADSIC_ADSITDirect	1362.118	126.956	317.371	89.380	55.637	0.000
ITDirect	1365.074	127.033	320.328	95.888	44.170	0.000
ADSITIC_ADSICDirect	1369.845	132.826	325.099	97.323	54.381	0.000
nullDirect	1370.323	133.121	325.577	102.532	46.392	0.000
ADSIT_ADSITDirect	1371.846	136.213	327.100	99.520	59.273	0.000
ADSITIC_ADSIT_ITIC_ADSICDirect	1372.257	134.925	327.511	98.383	61.847	0.000
ADSDirect	1372.713	131.716	327.966	100.481	50.297	0.000
ADSITDirect	1378.450	134.977	333.703	103.890	54.225	0.000
ADSICDirect	1384.285	138.312	339.538	106.092	55.895	0.000
ADSITICDirect	1385.248	140.784	340.502	108.633	58.695	0.000



Let's build a Hamiltonian Monte Carlo with the same formula:

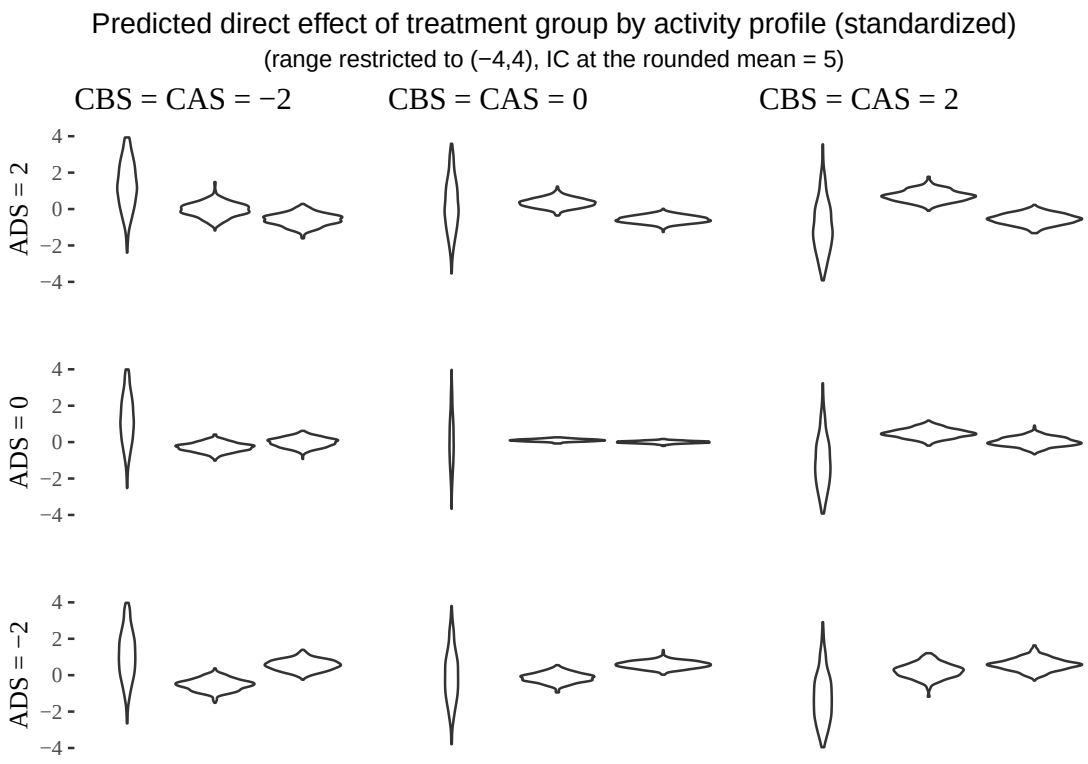
```
# tooFarDirectHMC <- ulam(
#   alist(
#     AdiffS ~ dnorm( mu, sigma ),
#     mu <- a + bADS[groupID] * ADS +  bIT[groupID] + bIC[groupID] * IC +
#       bADSIC * ADS * IC+ bCBS[groupID] *CBS + bCBSIC * CBS * IC + bCAS * CAS,
#     a ~ dnorm (0,0.3),
#     bADS[groupID] ~ dnorm(0,.3),
#     bADSIC ~ dnorm(0,.3),
#     bCAS ~ dnorm(0,0.3),
#     bCBS[groupID] ~ dnorm(0,.3),
#     bIT[groupID] ~ dnorm(0,.3),
#     bIC[groupID] ~ dnorm(0,.3),
#     bCBSIC ~ dnorm(0, .3),
#     sigma  ~ dexp(1)
#   ),
```

```
#   data = summaries
# )


#saveRDS(tooFarDirectHMC, file = "models/tooFarDirectHMC.rds")

tooFarDirectHMC <- readRDS(file = "models/tooFarDirectHMC.rds")
```

For a big picture, let's look at predicted direct effect by group and user activity profile.

```
visGroupDirect <- function (model, ADS, CBS, CAS, xmin =2, ymax = -3)
{
  groupID <- 1:3
  IC <- 5
  data <- expand.grid(ADS = ADS,groupID = groupID, CBS = CBS, CAS = CAS, IC =  IC)
  posterior <- extract.samples(model, n = 1e5)
  mu <- link( model, data=data )
  colnames(mu) <- levels(summaries$group)
  muLong <- melt(mu)
  colnames(muLong) <- c("id", "group", "AdiffS")
  means <-  round(apply(mu , 2 , mean ), 2)
  mu_HPDI <- round(apply( mu , 2 , HPDI ),2)
  means <- as.data.frame(means)
  means$group <- rownames(means)
  rownames(means) <- NULL
  meansDisp <- cbind(means,t(as.data.frame(mu_HPDI)))
  meansDisp <- meansDisp[,c(1,3,4)]

  plot <- ggplot(muLong)+geom_violin(aes(x = group, y = AdiffS), alpha = 0.2)+
    xlab("")+
    labs(title = paste("ADS=", ADS, ", CBS=",  CBS, ", CAS=", CAS,  sep = ""))+
    theme_tufte()+ylim(c(-4,4))
  #+   annotation_custom(tableGrob(meansDisp), xmin=xmin,  ymax=ymax)
  return(plot)
}



visGroupDirect2_2_2 <-  visGroupDirect(model = tooFarDirectHMC, ADS = 2,CBS = -2, CAS = -2)
visGroupDirect200 <-  visGroupDirect(model = tooFarDirectHMC, ADS = 2,CBS = 0, CAS = 0)
visGroupDirect222 <- visGroupDirect(model = tooFarDirectHMC, ADS = 2,CBS = 2, CAS = 2)

visGroupDirect0_2_2 <-  visGroupDirect(model = tooFarDirectHMC, ADS = 0,CBS = -2, CAS = -2)
visGroupDirect000 <-  visGroupDirect(model = tooFarDirectHMC, ADS = 0,CBS = 0, CAS = 0)
visGroupDirect022 <- visGroupDirect(model = tooFarDirectHMC, ADS = 0,CBS = 2, CAS = 2)


visGroupDirect_2_2_2 <-  visGroupDirect(model = tooFarDirectHMC, ADS = -2,CBS = -2, CAS = -2)
visGroupDirect_200 <-  visGroupDirect(model = tooFarDirectHMC, ADS = -2,CBS = 0, CAS = 0)
visGroupDirect_222 <- visGroupDirect(model = tooFarDirectHMC, ADS = -2,CBS = 2, CAS = 2)


visGroupDirectJoint <- ggarrange(visGroupDirect2_2_2+removeX +ylab("ADS = 2")+ggtitle("CBS = CAS = -2"),
                                 visGroupDirect200+removeX+removeY+ ggtitle("CBS = CAS = 0"),
                                 visGroupDirect222+removeX+removeY+ ggtitle("CBS = CAS = 2") ,
                         visGroupDirect0_2_2+removeX + ggtitle("")+ylab("ADS = 0"),
                                visGroupDirect000+removeX+removeY+ ggtitle(""),
                                visGroupDirect022+removeX+removeY+ ggtitle(""),
                         visGroupDirect_2_2_2+removeX+ylab("ADS = -2")+ ggtitle(""),
                                  visGroupDirect_200+removeX+removeY+ ggtitle(""),
                         visGroupDirect_222+removeX+removeY+ ggtitle("")
                                 ,ncol =3, nrow = 3)



visGroupDirectJoint2 <- annotate_figure(visGroupDirectJoint,
                                  top = text_grob("(range restricted to (-4,4), IC at the rounded mean =
                                                  size = 10))
visGroupJoint3 <- annotate_figure(visGroupDirectJoint2,
                                  top = text_grob("Predicted direct effect of treatment group by activit
                                                  size = 12))
```

```
visGroupJoint3
```



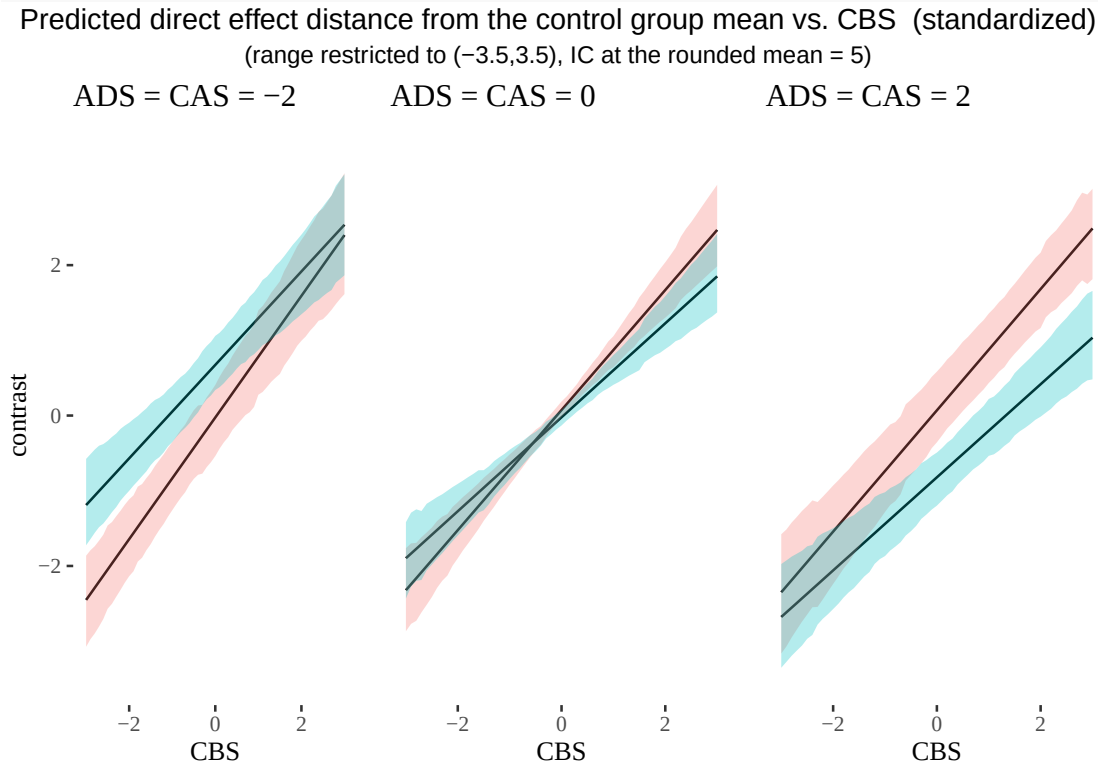
Now, predicted direct effect of five interventions vs. CBS, for three user activity profiles.

```
visContrastsCBSDirect <- function(model = FinalHMC, ADS = ADS , CAS= CAS, IC =  5,
                                  CBS = seq(-3,3,by  = 0.1)){
  groupID <- 1:3
  data <- expand.grid(ADS, groupID, CAS, IC , CBS)
  colnames(data) <- c("ADS", "groupID", "CAS", "IC", "CBS")
  posterior <- extract.samples(model, n = 1e5)
  link( model, data=data )
  mu <- link( model, data=data )

  means <-  round(apply(mu , 2 , mean ), 4)

  HPDIs <- round(apply( mu , 2 , HPDI ),4)
  visContrast <- cbind(data,means,t(as.data.frame(HPDIs)))

  ones <- 3 * (1:(nrow(visContrast)/3))-2
  twos <- 3 * (1:(nrow(visContrast)/3))-1
  threes <- 3 * (1:(nrow(visContrast)/3))

  colnames(visContrast)[c(7,8)] <- c("low", "high")

  contrast <- numeric(nrow(visContrast))
  cLow <- numeric(nrow(visContrast))
  cHigh <- numeric(nrow(visContrast))
  for(i in threes){
    contrast[i] <- visContrast$means[i] - visContrast$means[i-2]
  }
  for(i in twos){
    contrast[i] <- visContrast$means[i] - visContrast$means[i-1]
  }
  visContrast$contrast <- contrast
  visContrast$shift <-  visContrast$contrast - visContrast$means
  for(i in ones){
    visContrast$shift[i] <- 0
  }
  visContrast$cLow <- visContrast$low + visContrast$shift
  visContrast$cHigh <- visContrast$high + visContrast$shift
```

```
  visContrast$group = rep(c("control", "empathy", "normative"),
                          nrow(visContrast)/3)

  visContrastTreatment <- visContrast[groupID !=1,]

  return(ggplot(visContrastTreatment, aes(x = CBS, y = contrast, fill = group ))+
           geom_line(se = FALSE)+
           geom_ribbon(mapping =
                         aes(ymin = cLow, ymax = cHigh),
                       alpha = .3)+
           theme_tufte()+ylim(c(-3.5,3.5)))
}

visContrastDirect <-  ggarrange(visContrastsCBSDirect(model = tooFarDirect,ADS = -2, CAS = -2, IC = 5)+g
          visContrastsCBSDirect(model = tooFarDirect,ADS = 0, CAS = 0, IC = 5) +ggtitle("ADS = CAS = 0")
          visContrastsCBSDirect(model = tooFarDirect,ADS = 2, CAS = 2, IC = 5) +ggtitle("ADS = CAS = 2")
          ,ncol =3)
```

```
## Warning: Ignoring unknown parameters: se

## Warning: Ignoring unknown parameters: se

## Warning: Ignoring unknown parameters: se
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```

```
visContrastDirect2 <- annotate_figure(visContrastDirect,
                                      top = text_grob("(range restricted to (-3.5,3.5), IC at the roun
                                                      size = 10))
visContrastDirect3 <- annotate_figure(visContrastDirect2,
                                      top = text_grob("Predicted direct effect distance from the contr
                                                      size = 12))

visContrastDirect3
```



Now contrasts against ADS, for three user profiles:

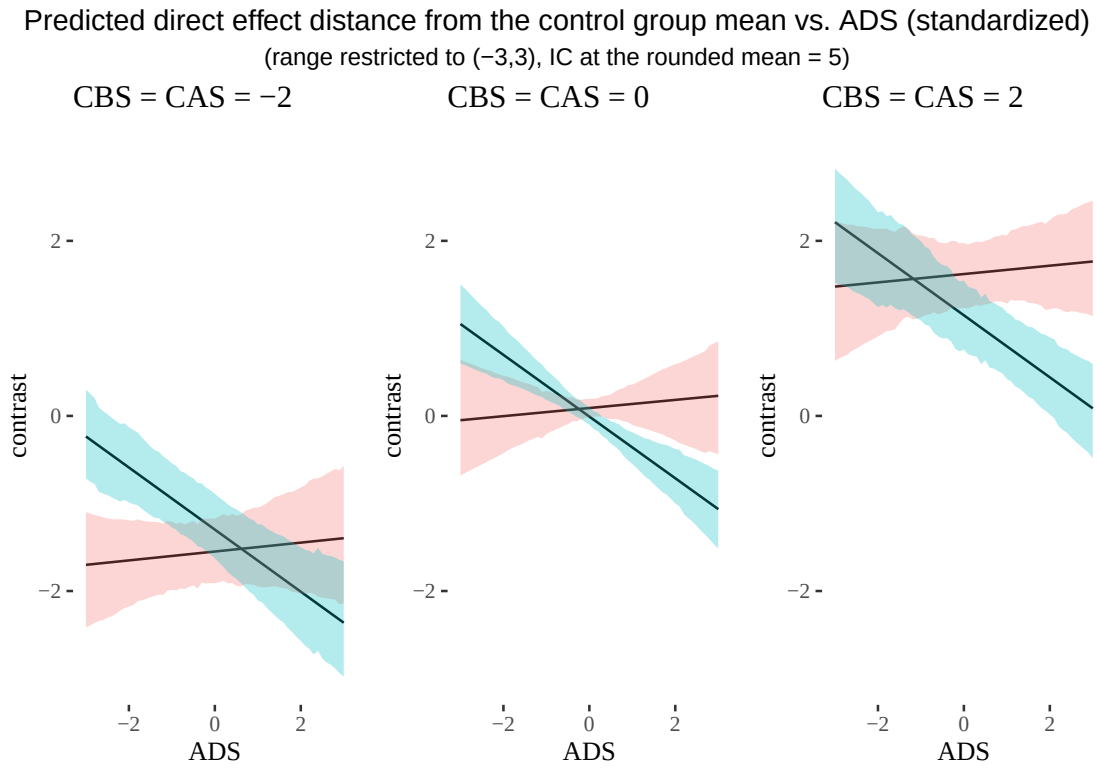
```
visContrastsADSDirect <- function(model = FinalHMC, CBS = CBS , CAS = CAS, IC =  5,
                          ADS = seq(-3,3,by  = 0.1))
{
  data <- expand.grid(CBS, groupID, CAS, IC , ADS)
  colnames(data) <- c("CBS", "groupID", "CAS", "IC", "ADS")
  posterior <- extract.samples(model, n = 1e5)
  mu <- link( model, data=data )
  means <-  round(apply(mu , 2 , mean ), 4)
  HPDIs <- round(apply( mu , 2 , HPDI ),4)
  visContrastADS <- cbind(data,means,t(as.data.frame(HPDIs)))


  ones <- 3 * (1:(nrow(visContrastADS)/3))-2
  twos <- 3 * (1:(nrow(visContrastADS)/3))-1
  threes <- 3 * (1:(nrow(visContrastADS)/3))

  colnames(visContrastADS)[c(7,8)] <- c("low", "high")
  contrastADS <- numeric(nrow(visContrastADS))
  for(i in threes){
    contrastADS[i] <- visContrastADS$means[i] - visContrastADS$means[i-2]
  }
  for(i in twos){
    contrastADS[i] <- visContrastADS$means[i] - visContrastADS$means[i-1]
  }
  visContrastADS$contrast <- contrastADS
  visContrastADS$shift <-  visContrastADS$contrast - visContrastADS$means
  for(i in ones){
    visContrastADS$shift[i] <- 0
  }
  visContrastADS$cLow <- visContrastADS$low + visContrastADS$shift
  visContrastADS$cHigh <- visContrastADS$high + visContrastADS$shift

  visContrastADS$group = rep(c("control", "empathy", "normative"),
                             nrow(visContrastADS)/3)
  visContrastTreatmentADS <- visContrastADS[groupID !=1,]

  return(ggplot(visContrastTreatmentADS, aes(x = ADS, y = contrast, fill = group ))+
           geom_line(se = FALSE) +
           geom_ribbon(mapping = aes(ymin = cLow, ymax = cHigh),
                       alpha = .3) +theme_tufte())
}
```

```
visContrastADSDirectJoint <- ggarrange(
  visContrastsADSDirect(tooFarDirect,CBS = -2, CAS = -2)+ggtitle("CBS = CAS = -2")+ylim(c(-3,3))+ scale_
  visContrastsADSDirect(tooFarDirect, CBS = 0, CAS = 0)+ggtitle("CBS = CAS = 0")+ylim(c(-3,3))+ scale_fi
  visContrastsADSDirect(tooFarDirect, CBS = 2, CAS = 2)+ggtitle("CBS = CAS = 2")+ylim(c(-3,3))+ scale_fi
  ncol =3)
```

```
## Warning: Ignoring unknown parameters: se

## Warning: Ignoring unknown parameters: se

## Warning: Ignoring unknown parameters: se
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```

```
visContrastADSDirectJoint2 <- annotate_figure(visContrastADSDirectJoint,
                                      top = text_grob("(range restricted to (-3,3), IC at the rounded
                                                      size = 10))
visContrastADSDirectJoint3 <- annotate_figure(visContrastADSDirectJoint2,
                                      top = text_grob("Predicted direct effect distance from the contr
                                                      size = 12))
```

```
visContrastADSDirectJoint3
```



References