DGE_EEG_exp_behavioral_analyses

Davide Gheza

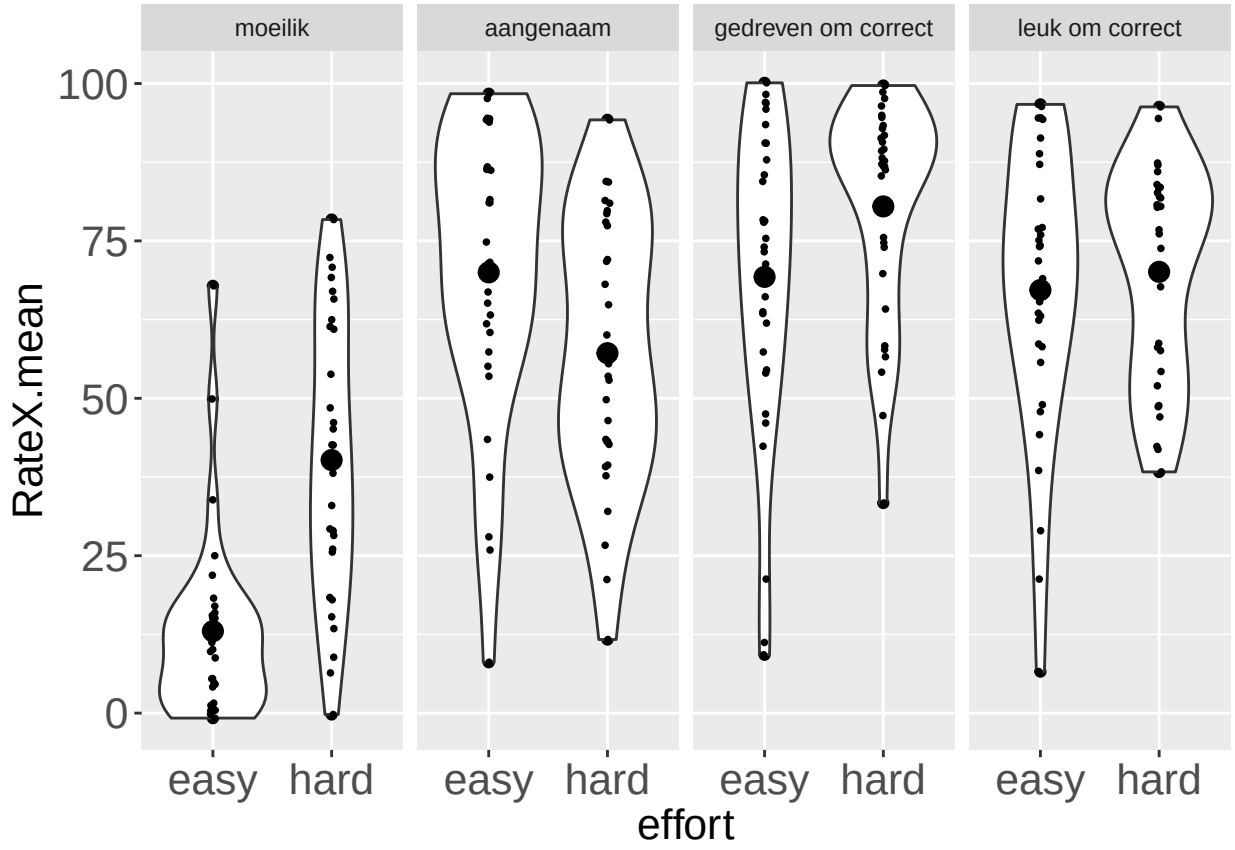
14 november 2018

```
## Doors.RESP - Door selection

# filter single entries of Doors.RESP
expdata.doorresp = expdata[!(expdata$SubTrial %in% c(2:10)),]

# set up data frame for by-group processing
doorresp.subj = group_by(expdata.doorresp, Subject, Doors.RESP)

# calculate the summary metrics: count of door selection by sbj and door
doorresp.subj = summarise(doorresp.subj,
                    doorresp.count = n()
                    )
# wide format
doorresp.subj = cast(doorresp.subj, Subject ~ Doors.RESP, value = "doorresp.count")
doorresp.subj
```

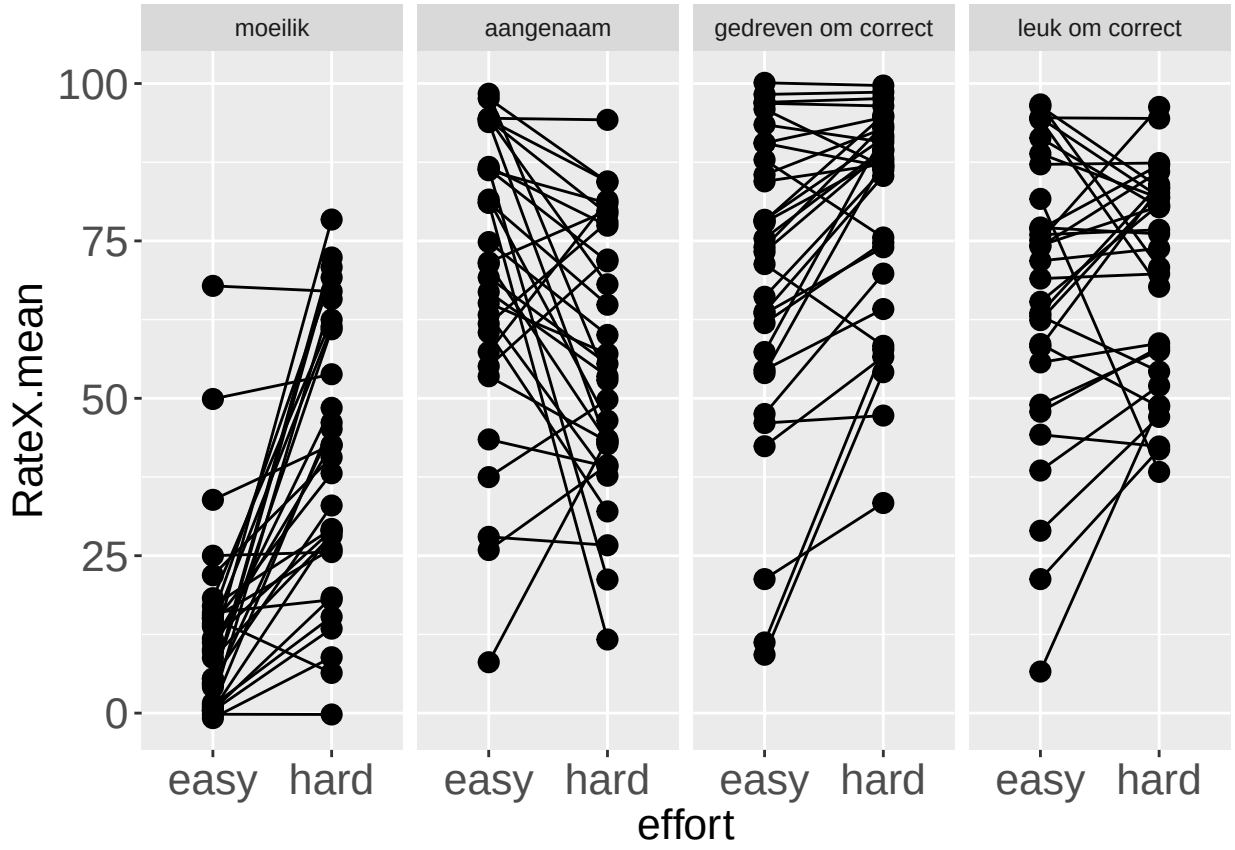
```
##    Subject  1   2  3   4
## 1        1 84  24 20  96
## 2        2 63  58 49  54
## 3        3 41  65 61  57
## 4        4 70  58 43  53
## 5        5 41  69 50  64
## 6        6 42  92 52  38
## 7        7 24  59 93  48
## 8        8 40  46 66  72
## 9        9 51  68 54  51
## 10      10 50  72 47  55
## 11      11 53  60 55  56
## 12      12 48  82 40  54
## 13      13 16  67 90  51
## 14      14 46  62 65  51
## 15      15 70  56 39  59
## 16      16 27  80 59  58
## 17      17 53  55 62  54
## 18      18 63  56 58  47
## 19      19 69  46 47  62
## 20      20 65  63 52  44
## 21      21 42  68 62  52
## 22      22 48  62 57  57
## 23      23 74  45 54  51
## 24      24 35 113 48  28
## 25      25 47  83 35  59
## 26      26 27  46 21 130
## 27      27 37  67 66  54
## 28      28 66  55 56  47
## 29      29 40  92 59  33
## 30      30 40  69 63  52
```

```
## 31      31 86  48 37  53
```

```
## RateX - Effort Task ratings

# filter RatingList
expdata.ETR = expdata[(expdata$Running.SubTrial. %in% "RatingList"),]

# compute VAS as percentage (relative to pixel range)
expdata.ETR$RateX = (expdata.ETR$RateX-316)/396*100

# set up data frame for by-group processing (RatingList = list of questions rated)
ETR.subj = group_by(expdata.ETR, Subject, RatingList)

# calculate the summary metrics: average of RateX by sbj and question
ETR.subj = summarise(ETR.subj,
                         RateX.mean = mean(RateX)
                     )
# wide format
ETR.wide = cast(ETR.subj, Subject ~ RatingList, value = "RateX.mean")


# plot ETR

# whole question: labels=expdata.ETR$text[1:8]

# create variables defining question type and effort level
ETR.subj$qtype = rep(c("moeilik", "aangenaam", "gedreven om correct", "leuk om correct"), each = 2)
ETR.subj$qtype = factor(ETR.subj$qtype,                                                                    # .
                        levels = c("moeilik", "aangenaam", "gedreven om correct", "leuk om correct"))
ETR.subj$effort = rep(c("easy", "hard"))

# plot violin
ETR.subj %>%
  ggplot(aes(x = effort, y = RateX.mean)) +
  geom_point() +
  geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
              position = "dodge", draw_quantiles = NULL, trim = TRUE,
              scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
  stat_summary(fun.y=mean, geom="point", shape=20, size=5) +
  geom_jitter(shape=16, position=position_jitter(0.02), size=1) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16)) +
  facet_grid(. ~ qtype)
```

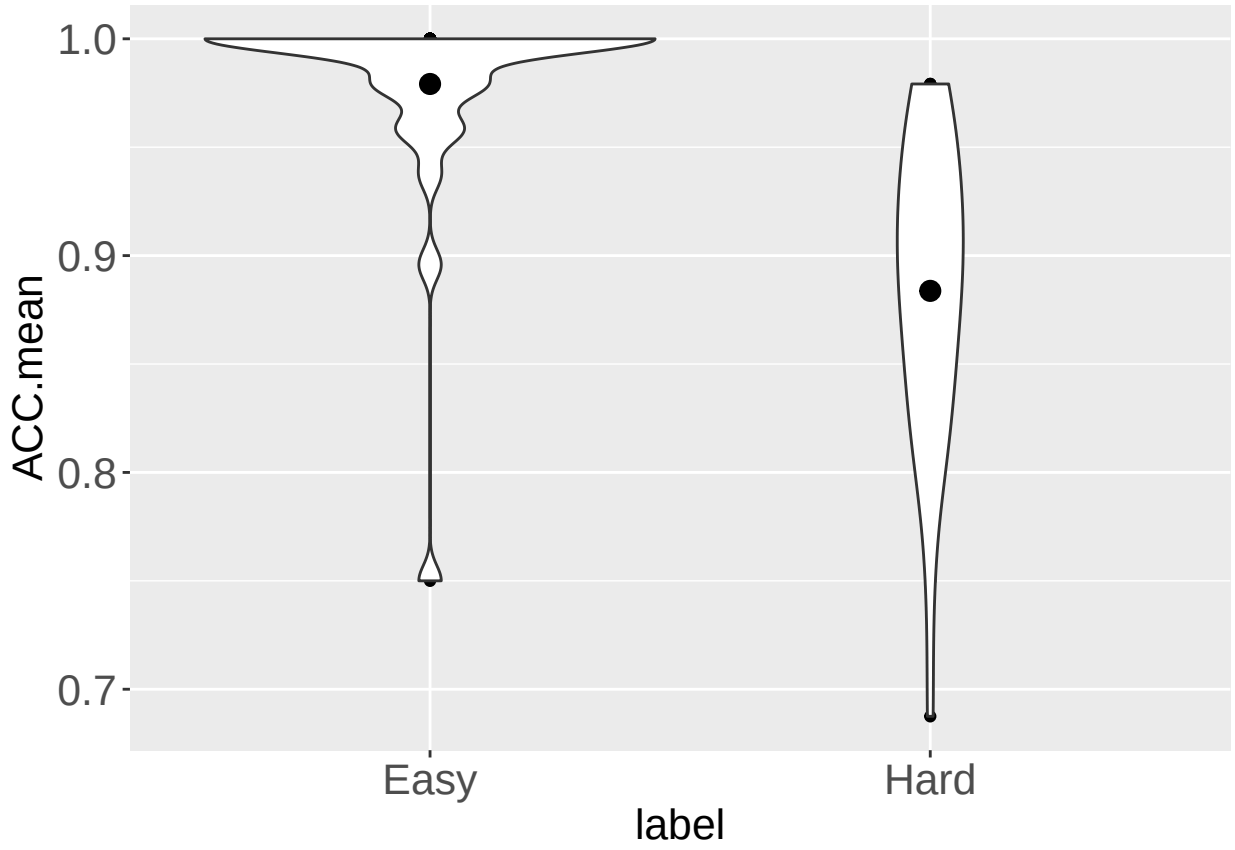



```
# plot geom line between effort levels for each sbj
ETR.subj %>%
  ggplot(aes(x = effort, y = RateX.mean, group = Subject)) +
  geom_point() +
  stat_summary(fun.y=mean, geom="point", shape=20, size=5) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16)) +
  geom_line() +
  facet_grid(. ~ qtype)
```




```
## EffortResponse.ACC

#filter EffortTaskList

expdata.acc = expdata[(expdata$Running.SubTrial. %in% "EffortTaskList"),]

# sort by sbj and cond
expdata.acc = expdata.acc[order(expdata.acc$Subject, expdata.acc$Condition),]

# subset
expdata.acc = subset(expdata.acc, select = c(Subject, Condition, EffortResponse.ACC))

# set up data frame for by-group processing
expdata.acc = group_by(expdata.acc, Condition)


# calculate the summary metrics
acc.sum = summarise(expdata.acc,
#                     acc.count = count(EffortResponse.ACC),
                     acc.mean = mean(EffortResponse.ACC))


# plot accuracy with prob density

expdata.acc = expdata.acc[!(expdata.acc$Condition %in% c("32", "52")),] # filter out Condition = 32 52
```

```r
acc.subj = group_by(expdata.acc, Condition, Subject) # set up data frame for by-group processing

acc.subj = summarise(acc.subj,                       # calculate the summary metrics - mean for Subject*
                     ACC.mean = mean(EffortResponse.ACC))

# plot expdata.acc

acc.subj$Condition = as.factor(acc.subj$Condition)

acc.subj$label = factor(acc.subj$Condition,
                        labels=c("Easy", "Hard"))


acc.subj %>%
  ggplot(aes(x = label, y = ACC.mean)) +
  geom_point() +
  geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
              position = "dodge", draw_quantiles = NULL, trim = TRUE,
              scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
  stat_summary(fun.y=mean, geom="point", shape=20, size=5) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16))
```

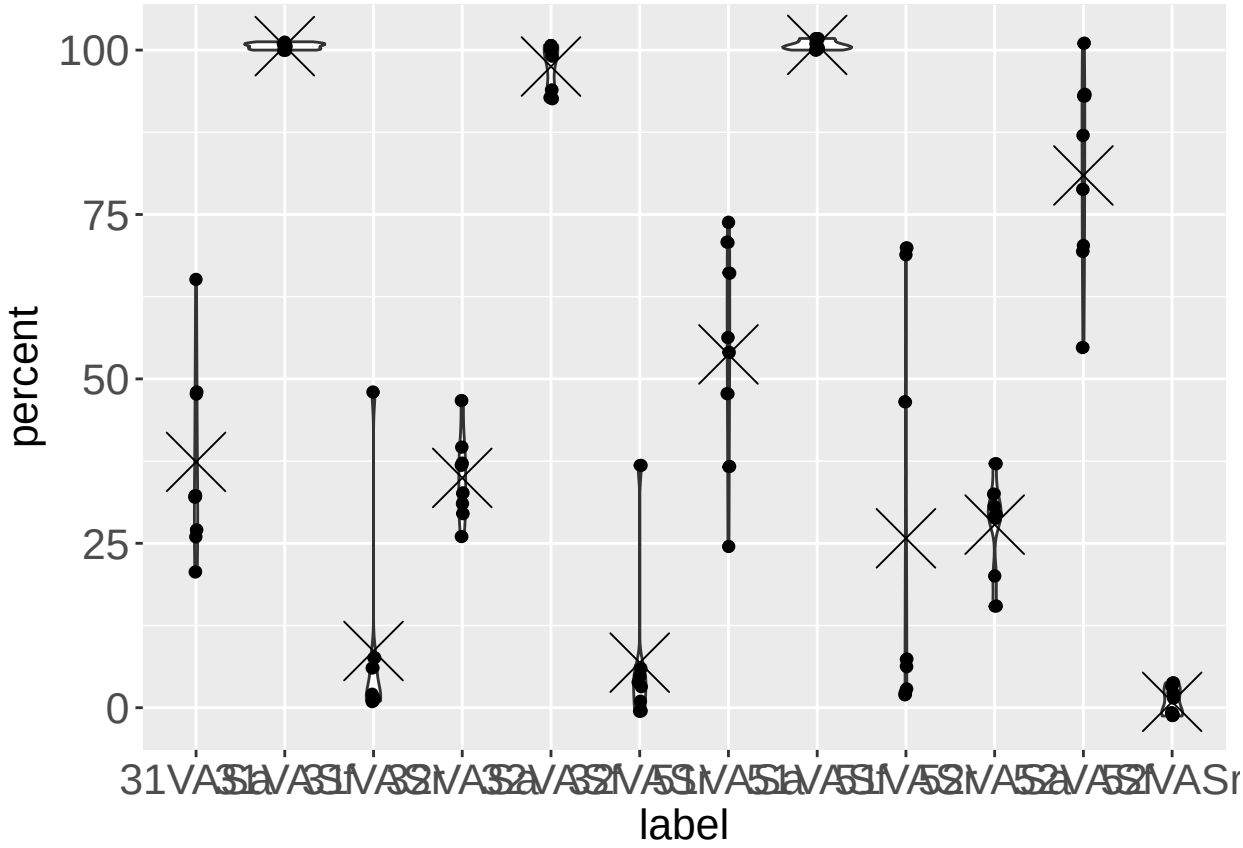



```
## EffortResponse.RT
```

```r
#filter EffortTaskList

expdata.RT = expdata[(expdata$Running.SubTrial. %in% "EffortTaskList"),]

# filter EffortResponse.ACC != 0 (exclude mistakes)

expdata.RT = expdata.RT[!(expdata.RT$EffortResponse.ACC %in% "0"),]


# sort by sbj and cond
expdata.RT = expdata.RT[order(expdata.RT$Subject, expdata.RT$Condition),]

# subset
expdata.RT = subset(expdata.RT, select = c(Subject, Condition, EffortResponse.RT))

# filter out Condition = 32 52 (exclude conditions where effort task followed the rating)
expdata.RT = expdata.RT[!(expdata.RT$Condition %in% c("32", "52")),]

# set up data frame for by-group processing
RT.subj = group_by(expdata.RT, Condition, Subject)

# calculate the summary metrics - mean for Subject*Condition
RT.subj = summarise(RT.subj,
                        RT.mean = mean(EffortResponse.RT))

# calculate the summary metrics - mean for Condition (on within Subject average)
RT.sum = group_by(RT.subj, Condition)
RT.sum = summarise(RT.sum,
                   RT.mean = mean(RT.mean))


# plot RT

RT.subj$Condition = as.factor(RT.subj$Condition)

RT.subj$label = factor(RT.subj$Condition,
                labels=c("Easy", "Hard"))  # excluded:  "EasyRating" , "HardRating"


RT.subj %>%
  ggplot(aes(x = label, y = RT.mean)) +
  geom_point() +
  geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
            position = "dodge", draw_quantiles = NULL, trim = TRUE,
            scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
  stat_summary(fun.y=mean, geom="point", shape=15, size=5, color="red") +
  geom_jitter(shape=16, position=position_jitter(0.02), size=2) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16))
```

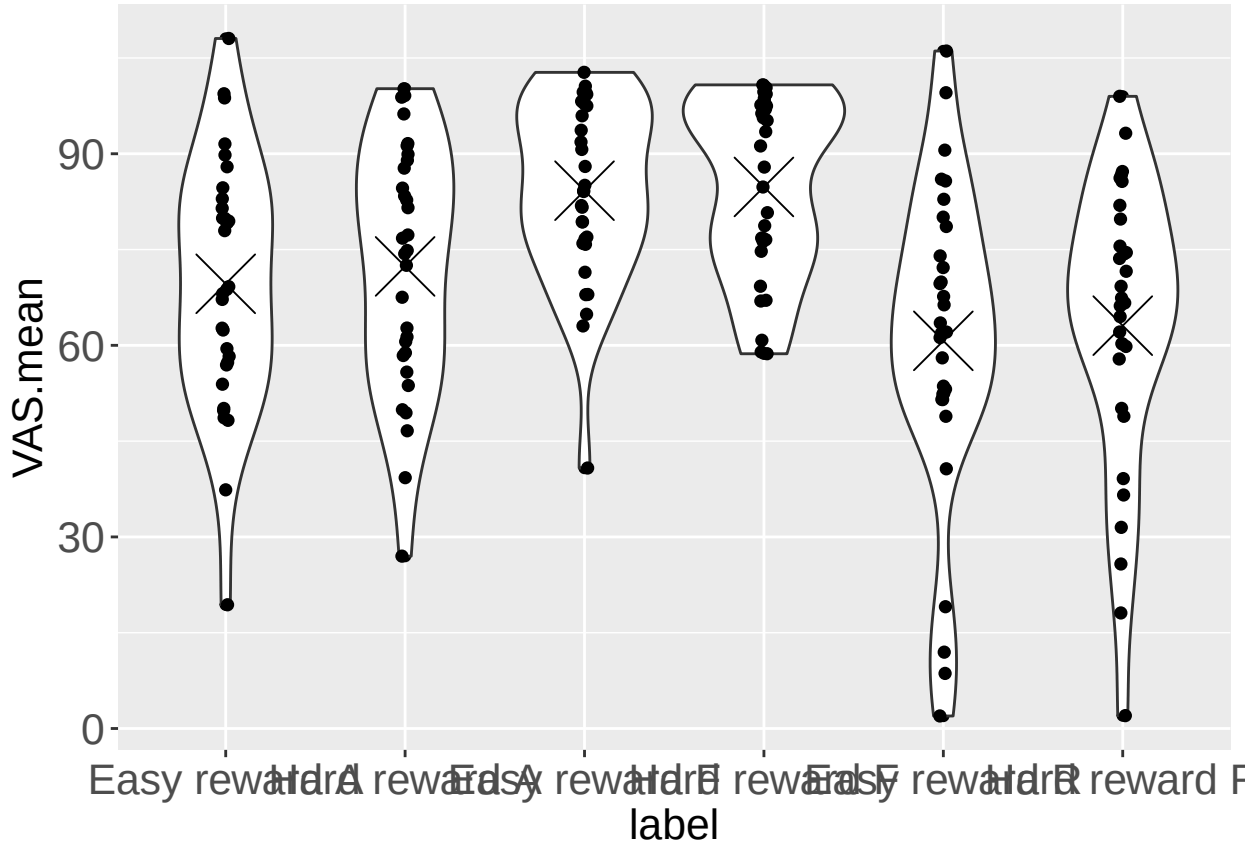

```
## FB ratings

#filter FbRateProc

expdata.FbRate = expdata[(expdata$Procedure.SubTrial. %in% "FbRateProc"),]

# sort by sbj and cond
expdata.FbRate = expdata.FbRate[order(expdata.FbRate$Subject, expdata.FbRate$Condition),]

# subset
expdata.FbRate = subset(expdata.FbRate, select = c(Subject, Condition, FbRateXa, FbRateXf, FbRateXr))

# add FBrate repetition n. (note: 12 rating in behavioral exp. only 8 in EEG exp)

expdata.FbRate$rateRep = rep((1:8), times = ((length(unique(expdata.FbRate$Subject)))*
                                               (length(unique(expdata.FbRate$Condition))))
                             )

# compute VAS as percentage (relative to pixel range)
expdata.FbRate$VASa = (expdata.FbRate$FbRateXa-316)/396*100
expdata.FbRate$VASf = (expdata.FbRate$FbRateXf-316)/396*100
expdata.FbRate$VASr = (expdata.FbRate$FbRateXr-316)/396*100

# invert score for Frustrerend

expdata.FbRate$VASf = 100-expdata.FbRate$VASf
```

```r
# long format: RateType as condition

expdata.FbRate = gather(expdata.FbRate, key = "RateType", value = "percent", VASa:VASr)
expdata.FbRate = subset(expdata.FbRate, select = c(Subject, Condition, rateRep, RateType, percent)) # d

# create factor outcome

expdata.FbRate$outcome[(expdata.FbRate$Condition %in% c("31","51"))] = "reward"
expdata.FbRate$outcome[(expdata.FbRate$Condition %in% c("32","52"))] = "noreward"

expdata.FbRate$outcome = as.factor(expdata.FbRate$outcome)

# create factor effortlevel

expdata.FbRate$efflev[(expdata.FbRate$Condition %in% c("31","32"))] = "low"
expdata.FbRate$efflev[(expdata.FbRate$Condition %in% c("51","52"))] = "high"

expdata.FbRate$efflev = as.factor(expdata.FbRate$efflev)




### inspect single subject ratings ###

expdata.FbRate %>%
  mutate(label = paste0(expdata.FbRate$Condition, expdata.FbRate$RateType)) %>%  # create label definin
  filter(Subject == 1) %>%                                                       # filter sbj n
  ggplot(aes(x = label, y = percent)) +
    geom_point() +
    geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
                position = "dodge", draw_quantiles = NULL, trim = TRUE,
                scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
    stat_summary(fun.y=mean, geom="point", shape=4, size=10, color="black") +
    geom_jitter(shape=16, position=position_jitter(0.02), size=2) +
    theme(axis.text = element_text(size = 16),
          axis.title = element_text(size = 16))
```

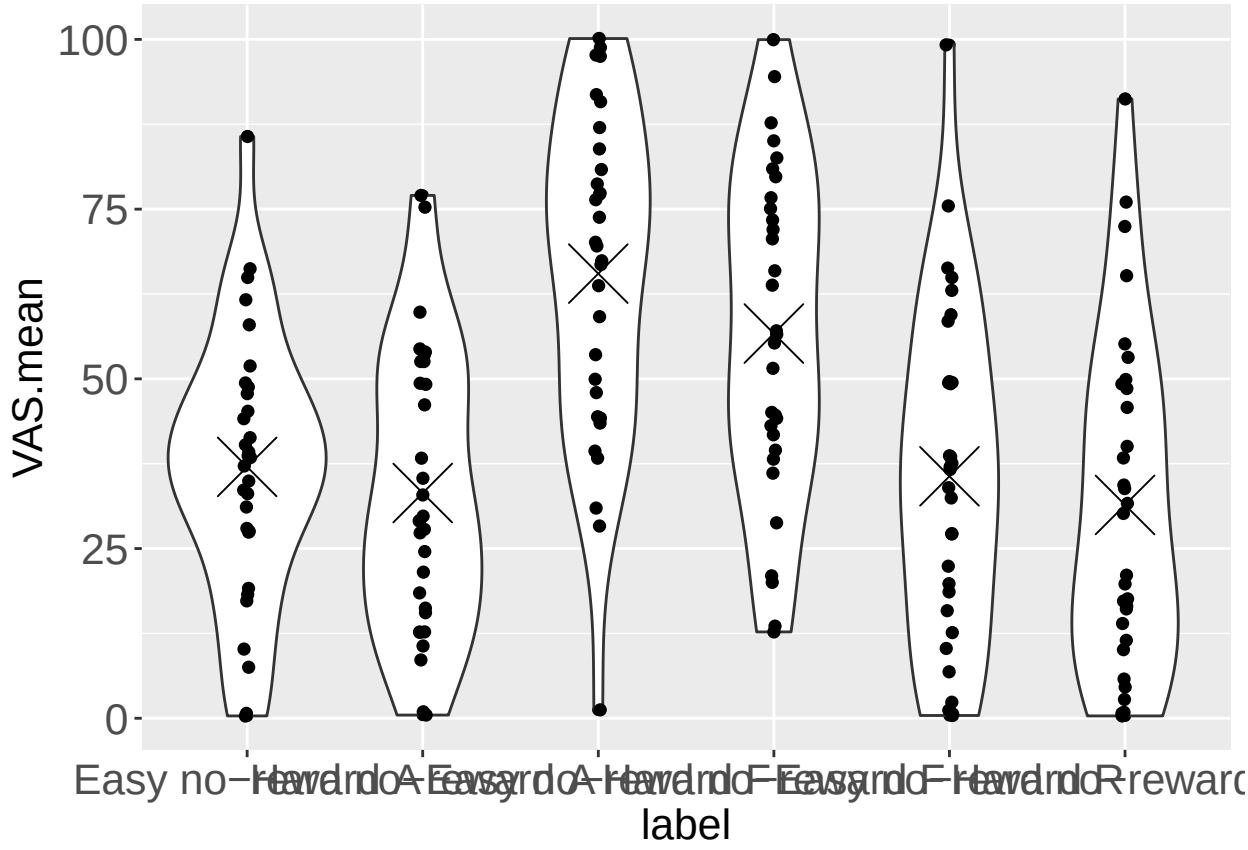



```
############## Exporting FB rating - sbj level AVERAGES by Condition and RateType #############

# set up data frame for by-group processing
FbRate.subj = group_by(expdata.FbRate, Condition, Subject, RateType)

# calculate the summary metrics - mean for Subject*Condition*RateType
FbRate.subj = summarise(FbRate.subj,
                        VAS.mean = mean(percent))

# save out expdata.FbRate (with inverted score for Frustrerend)
  # wide format
  FbRate.subj$widecond = paste0(FbRate.subj$Condition, FbRate.subj$RateType)
  FbRate.subj.wide = cast(FbRate.subj, Subject ~ widecond, value = "VAS.mean")
  # write.csv
  # write.csv(FbRate.subj.wide, file = "23_behavioral_FBrate_summarised_invertedFrustrerend.csv")

# log transform for non-normal distributed data

# ## set negative values as 0.1 (out of scale ratings)
# FbRate.subj$VAS.mean[FbRate.subj$VAS.mean<0] = 0.1
# ## log transform
# FbRate.subj$VAS.mean = log(FbRate.subj$VAS.mean,base= exp(10))


############## plotting FB rating - sbj level averages ##############
```

```r
FbRate.subj$Condition = as.factor(FbRate.subj$Condition)

FbRate.subj$RateType = as.factor(FbRate.subj$RateType)

FbRate.subj$label = paste0(FbRate.subj$Condition, FbRate.subj$RateType)

FbRate.subj$label = factor(FbRate.subj$label,
                       levels = c("31VASa", "51VASa", "31VASf", "51VASf", "31VASr", "51VASr","32VASa",
                       labels=c("Easy reward A", "Hard reward A", "Easy reward F", "Hard reward F", "Eas


# reward

FbRate.subj.reward = FbRate.subj[(FbRate.subj$Condition %in% c("31", "51")),]

FbRate.subj.reward = group_by(FbRate.subj.reward, Condition, Subject, RateType)

FbRate.subj.reward %>%
  ggplot(aes(x = label, y = VAS.mean)) +
  geom_point() +
  geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
              position = "dodge", draw_quantiles = NULL, trim = TRUE,
              scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
  stat_summary(fun.y=mean, geom="point", shape=4, size=10, color="black") +
  geom_jitter(shape=16, position=position_jitter(0.02), size=2) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16))
```




```
# no-reward

FbRate.subj.noreward = FbRate.subj[(FbRate.subj$Condition %in% c("32", "52")),]

FbRate.subj.noreward = group_by(FbRate.subj.noreward, Condition, Subject, RateType)

FbRate.subj.noreward %>%
  ggplot(aes(x = label, y = VAS.mean)) +
  geom_point() +
  geom_violin(mapping = NULL, data = NULL, stat = "ydensity",
              position = "dodge", draw_quantiles = NULL, trim = TRUE,
              scale = "area", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE) +
  stat_summary(fun.y=mean, geom="point", shape=4, size=10, color="black") +
  geom_jitter(shape=16, position=position_jitter(0.02), size=2) +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 16))
```



stats on FBratings (single rating level)

```
########## Bayesian model comparison ##############

library(BayesFactor)

num.iter=10000 # number of MonteCarlo iterations (default: 10000)

# as factor
expdata.FbRate$Subject = as.factor(expdata.FbRate$Subject)
expdata.FbRate$rateRep = as.factor(expdata.FbRate$rateRep)
expdata.FbRate$RateType = as.factor(expdata.FbRate$RateType)

## Specification of random effects
# 1) Subject*RateType = random intercept for Subject, random effect of RateType, random effect of RateT


# Assuming a medium Cauchy prior d~Cauchy(0,.707):
m.null=lmBF(percent ~ 1 + Subject*RateType,
            data=expdata.FbRate,iterations=num.iter,whichRandom=c("Subject*RateType"),
            rscaleRandom="nuisance",rscaleFixed=.707)
m.outcome=lmBF(percent ~ Subject*RateType + outcome,
               data=expdata.FbRate,iterations=num.iter,whichRandom=c("Subject*RateType"),
               rscaleRandom="nuisance",rscaleFixed=.707)
m.efflev=lmBF(percent ~ Subject*RateType + efflev,
```

```
              data=expdata.FbRate,iterations=num.iter,whichRandom=c("Subject*RateType"),
              rscaleRandom="nuisance",rscaleFixed=.707)
m.maineffects=lmBF(percent ~ Subject*RateType + outcome + efflev,
                   data=expdata.FbRate,iterations=num.iter,whichRandom=c("Subject*RateType"),
                   rscaleRandom="nuisance",rscaleFixed=.707)
m.interaction=lmBF(percent ~ Subject*RateType + outcome * efflev,
                   data=expdata.FbRate,iterations=num.iter,whichRandom=c("Subject*RateType"),
                   rscaleRandom="nuisance",rscaleFixed=.707)
```

BF model x | null

```
m.outcome/m.null
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome : 2.938238e+263 ±2.01%
##
## Against denominator:
##   percent ~ 1 + Subject * RateType
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.efflev/m.null
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + efflev : 0.2110546 ±2.13%
##
## Against denominator:
##   percent ~ 1 + Subject * RateType
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.maineffects/m.null
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome + efflev : 2.002846e+263 ±4.92%
##
## Against denominator:
##   percent ~ 1 + Subject * RateType
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.interaction/m.null
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome * efflev : 2.428436e+267 ±4.71%
##
## Against denominator:
##   percent ~ 1 + Subject * RateType
## ---
## Bayes factor type: BFlinearModel, JZS
```

BF model x | model y

```
m.interaction/m.maineffects
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome * efflev : 12124.93 ±6.55%
##
## Against denominator:
##   percent ~ Subject * RateType + outcome + efflev
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.interaction/m.efflev
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome * efflev : 1.150619e+268 ±4.82%
##
## Against denominator:
##   percent ~ Subject * RateType + efflev
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.interaction/m.outcome
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome * efflev : 8264.94 ±4.76%
##
## Against denominator:
##   percent ~ Subject * RateType + outcome
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
m.interaction/m.null
```

```
## Bayes factor analysis
## --------------
## [1] Subject * RateType + outcome * efflev : 2.428436e+267 ±4.71%
##
## Against denominator:
##   percent ~ 1 + Subject * RateType
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
# chains = posterior(m.interaction, iter=10000)
# summary(chains)

# plot(m.interaction, include1=FALSE, addDenom = FALSE)
# ?plot
```

–> the interaction model is the best one.

–> the interaction model is the best one. This interaction can be interpreted as "Participant rated the reward FB as more pleasant when they anticipated high vs low cognitive effort, while they rated the no-reward FB as more pleasant when they anticipated low vs high cognitive effort

–actually, for claims at each level or Reward, we need to first run a t-test within reward level...