

coAuthor-ENA

Zach

2023-08-30

Contents

Load packages	1
Prep data	1
Read data	2
Add metadata	2
Prep for ENA model	2
Run ENA accumulation	3
checking networks	3
View space	4
Statistical tests	5
Set up data and check data	5
Checking points	5
Checking other groups	6
Clustering of observations	12
Regression analysis	14
ENA plots	19
Mean networks (ownership)	19
Network subtraction (temp)	23

Load packages

```
rm(list=ls()) #clear environment

library(rENA)
#library(ona)
#library(tma)
library(tidyverse) #for wrangling
library(lmerTest) #for hlms
library(ICC) #for testing clustering of observations
library(emmeans) #for comparing subpopulations
library(performance) #for regression diagnostics
```

Prep data

Read data

```
# data1 <- read.csv('ena_all_with_groups.csv', stringsAsFactors = FALSE) #read data
#
# #read metadata
#
# meta_cr = read.csv("~/Rprojects/Yixin/CoAuthor - Metadata & Survey - Metadata (creative).csv",
#                   stringsAsFactors = FALSE)
#
# meta_arg = read.csv("~/Rprojects/Yixin/CoAuthor - Metadata & Survey - Metadata (argumentative).csv",
#                   stringsAsFactors = FALSE)
#
# meta_coauthor = bind_rows(meta_cr, meta_arg)

load("~/Rprojects/Yixin/accum-300823.Rdata")
```

Add metadata

```
#data1 = left_join(data1, meta_coauthor, by = c("worker_id", "session_id"))
```

Prep for ENA model

```
# units = data1[, c("session_id",
#                  "worker_id")]
#
#
#
# conversation = data1[, c("session_id",
#                          "worker_id",
#                          "sentSeq")]
#
# codeCols = c(
#   'compose',
#   #'delete',
#   'relocate',
#   'reflect',
#   'seekSugg',
#   'acceptSugg',
#   'dismissSugg',
#   'lowModification',
#   'highModification',
#   'reviseSugg',
#   'reviseUser'
# )
#
# codes = data1[, codeCols]
#
# #mask =
#
# meta = data1[, c("genre",
#                 "highTemp",
```

```
#           "ownershipMetadata",
#           "prompt_code"
#       )]
```

Run ENA accumulation

```
# accum =
#   ena.accumulate.data(
#     units = units,
#     conversation = conversation,
#     codes = codes,
#     metadata = meta,
#     #mask = mask,
#     window.size.back = "inf" # each line in the conversation can connect back to the first line--allows
# )

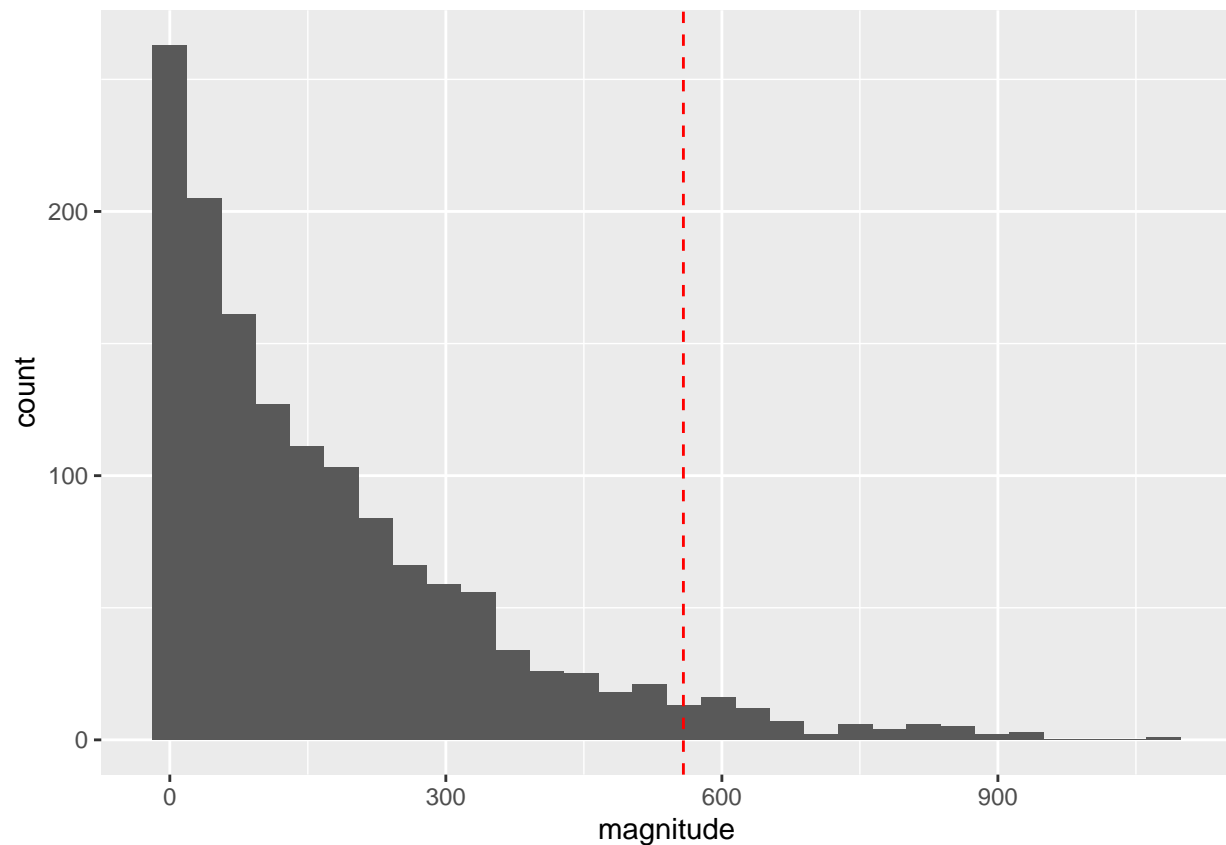
set = ena.make.set(accum)
```

checking networks

```
sparse.nets = accum$connection.counts %>% rowwise(ENA_UNIT) %>% mutate(magnitude = norm(c_across(contains("net")), method = "l2"))

x = quantile(sparse.nets$magnitude, 0.95)

ggplot(sparse.nets, aes(x = magnitude)) + geom_histogram() + geom_vline(xintercept = x, linetype = "dashed")
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
zeros = which(sparse.nets$magnitude == 0)
length(zeros)
#> [1] 24
```

```
#Run ENA dimensional reduction
```

```
set = ena.make.set(enadata = accum)
```

View space

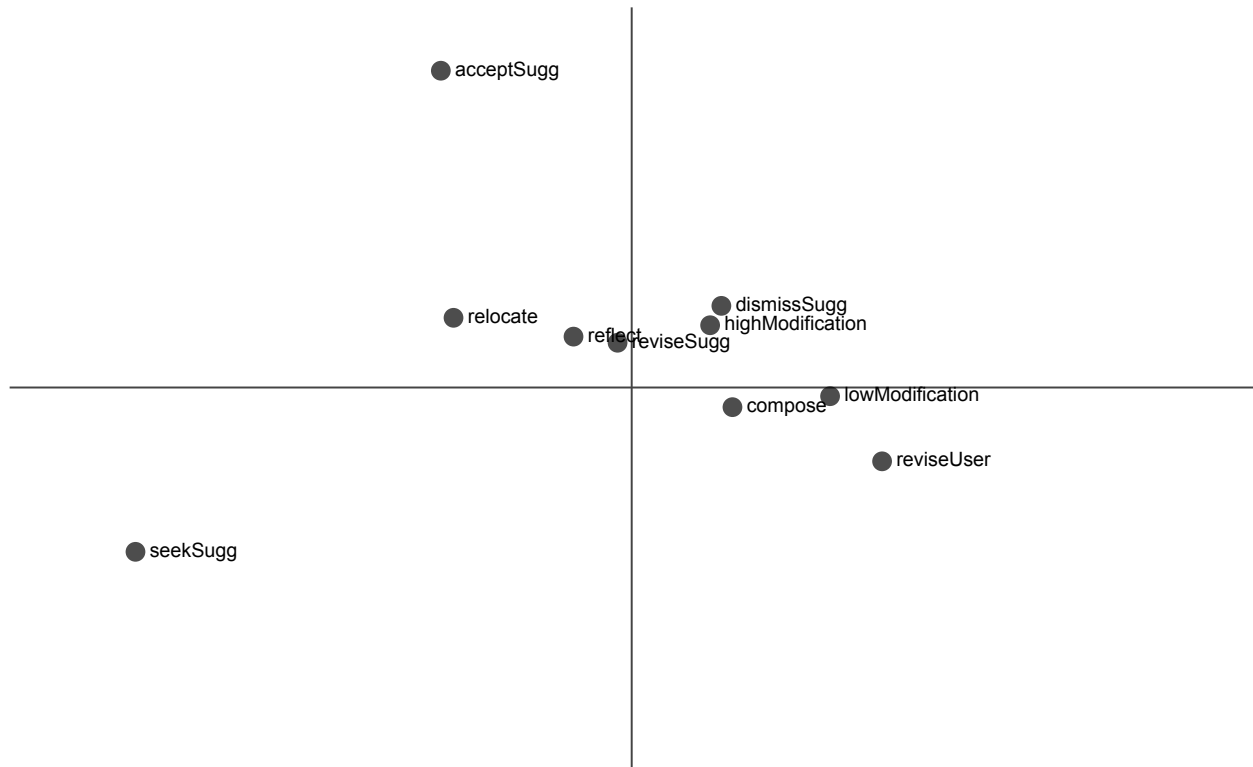
```
network = as.matrix(set$line.weights)
mean_network = colMeans(network)

network_mult = 0

p = ena.plot(set, title = "Overall Mean Network") %>%
  ena.plot.network(mean_network * network_mult, colors = "black")

p$plot
```

Overall Mean Network



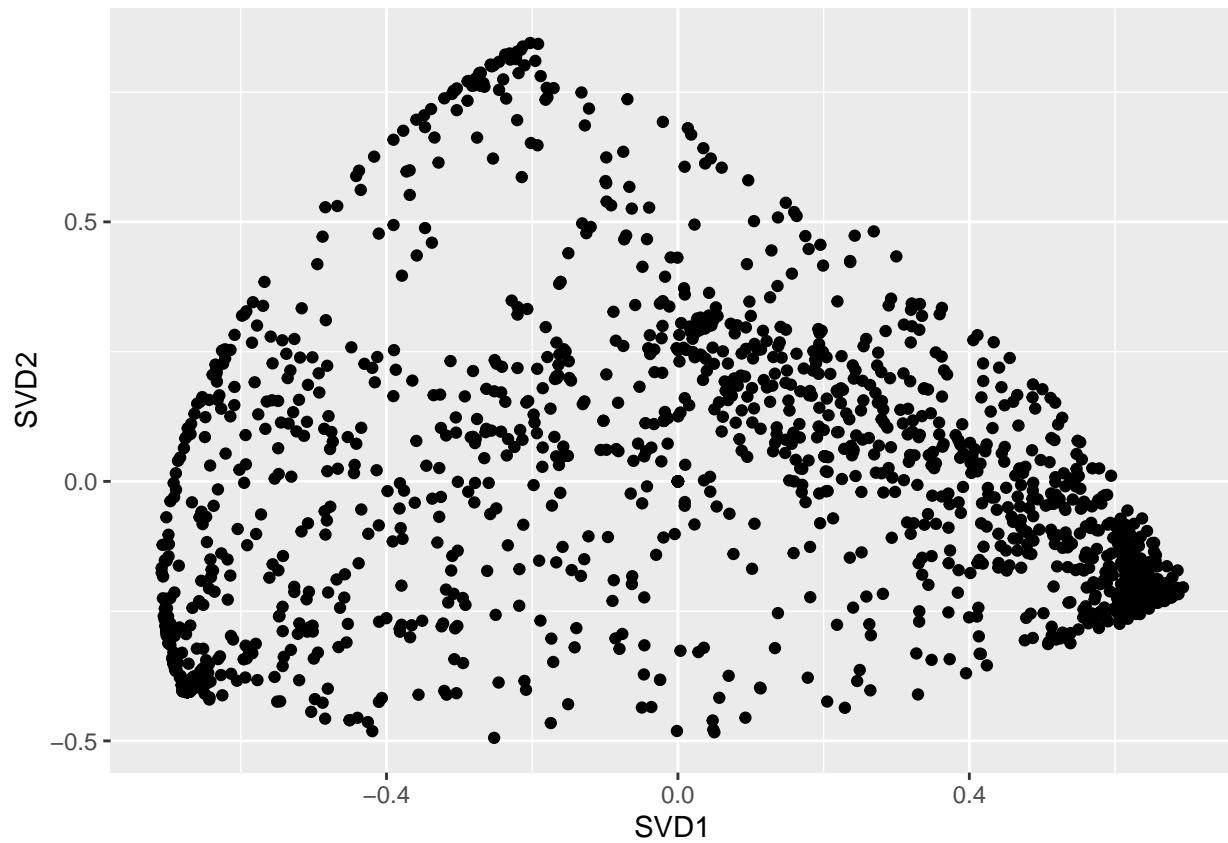
Statistical tests

Set up data and check data

```
#names(set$points)
reg_data = set$points[,c(1:9)]
#glimpse(reg_data)
#table(reg_data$genre)
#t(table(reg_data$genre, reg_data$worker_id))
#summary(reg_data)
```

Checking points

```
ggplot(reg_data, aes(x = SVD1, y = SVD2)) + geom_point()
```

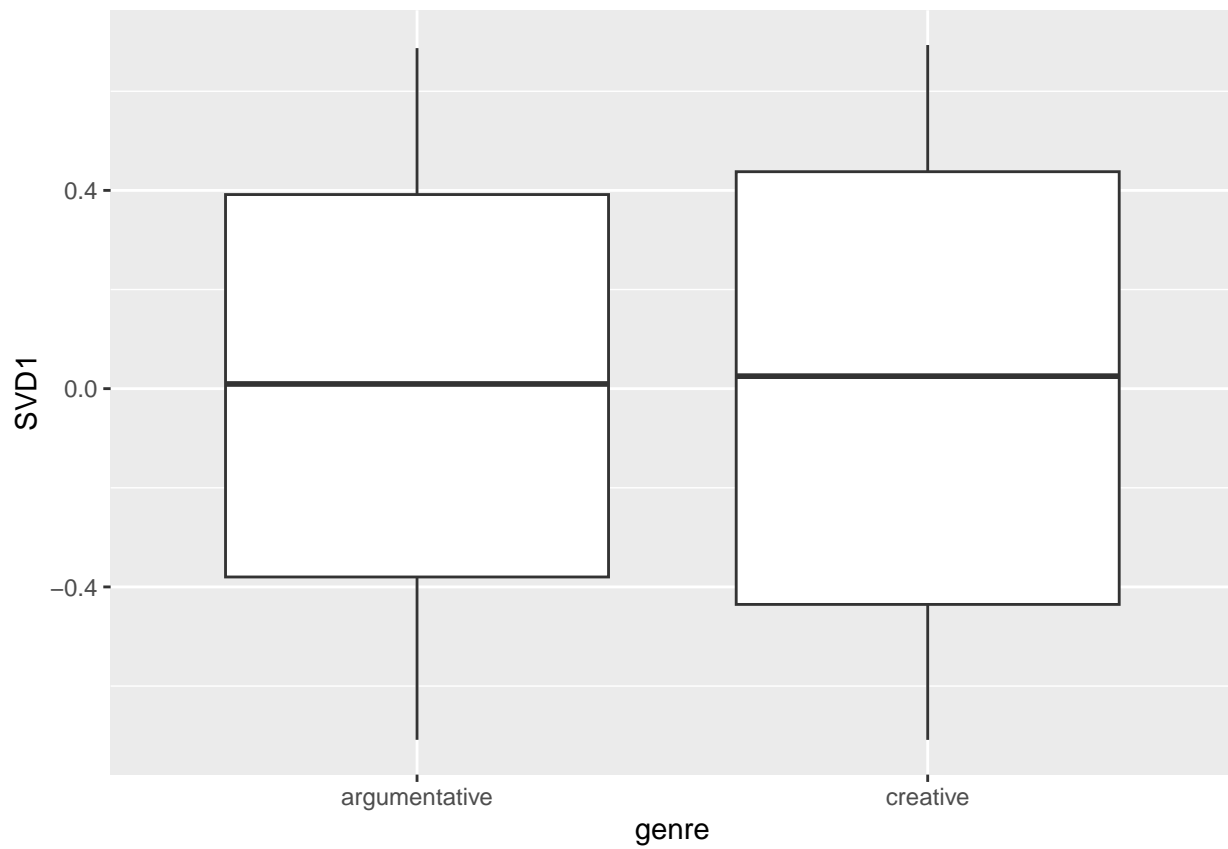


Fanning shape is strange to me

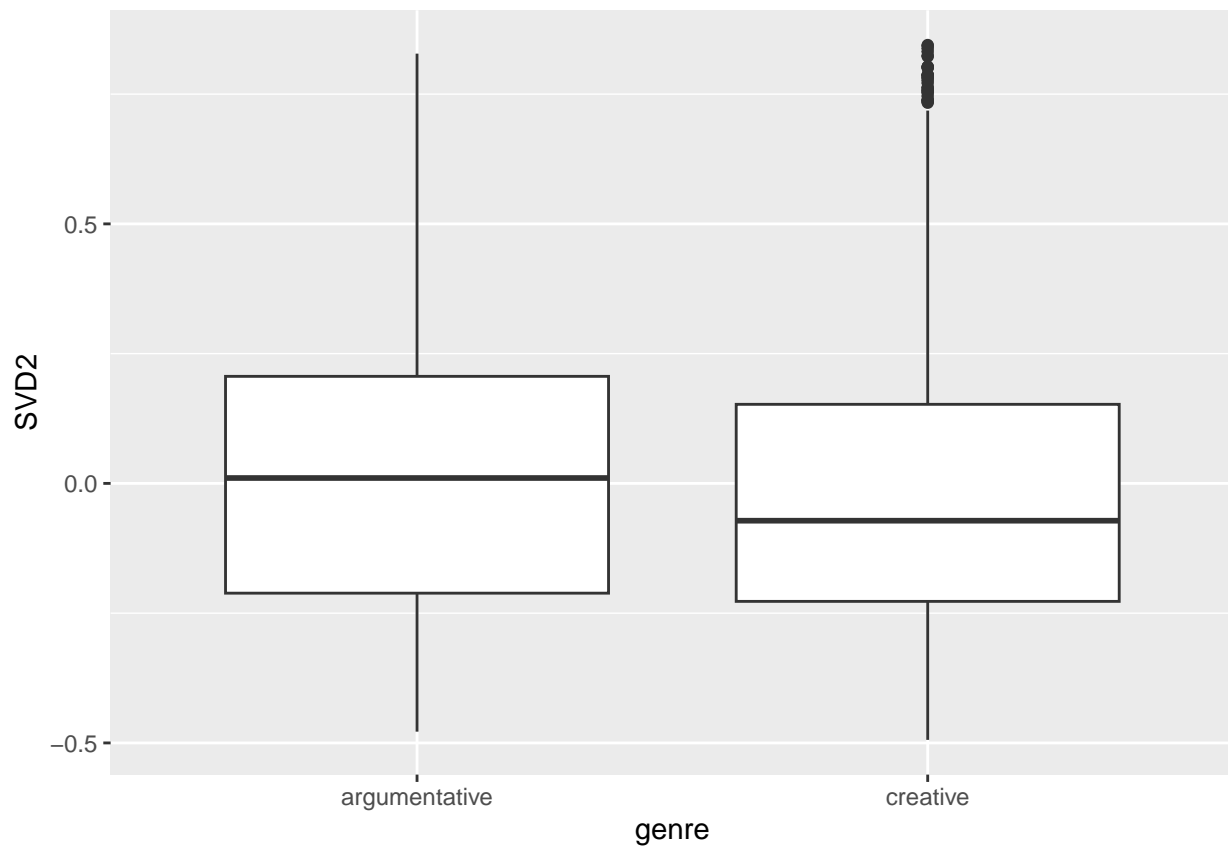
Checking other groups

genre

```
ggplot(reg_data, aes(x = genre, y = SVD1)) + geom_boxplot()
```

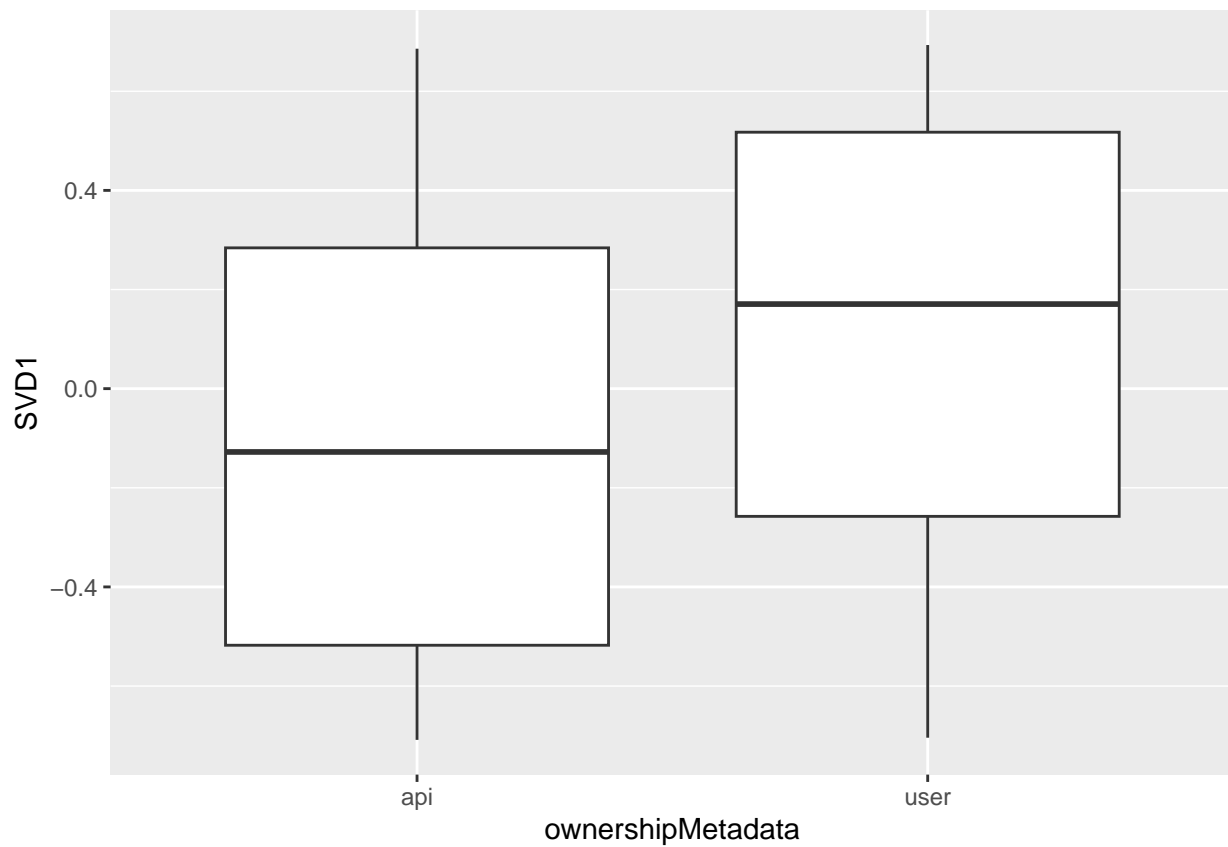


```
ggplot(reg_data, aes(x = genre, y = SVD2)) + geom_boxplot()
```

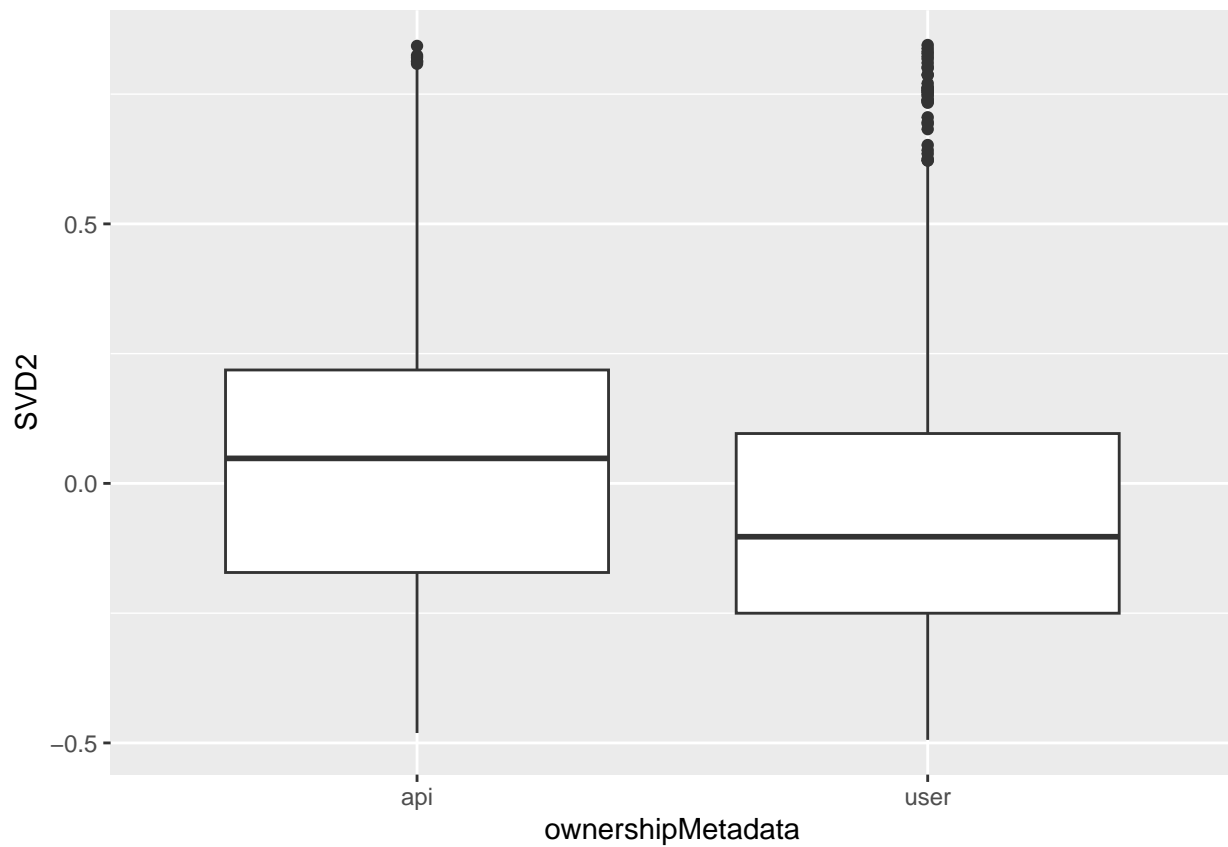


ownership

```
ggplot(reg_data, aes(x = ownershipMetadata, y = SVD1)) + geom_boxplot()
```

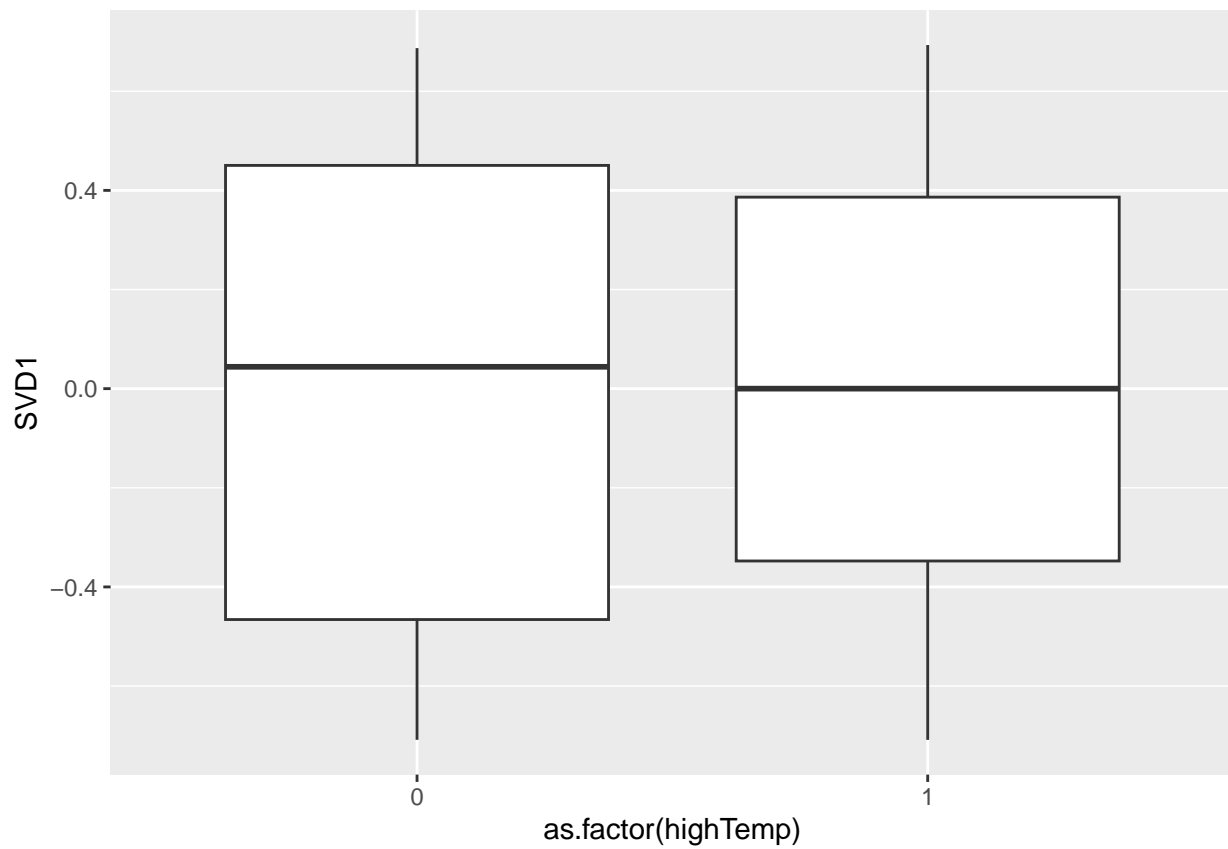



```
ggplot(reg_data, aes(x = ownershipMetadata, y = SVD2)) + geom_boxplot()
```

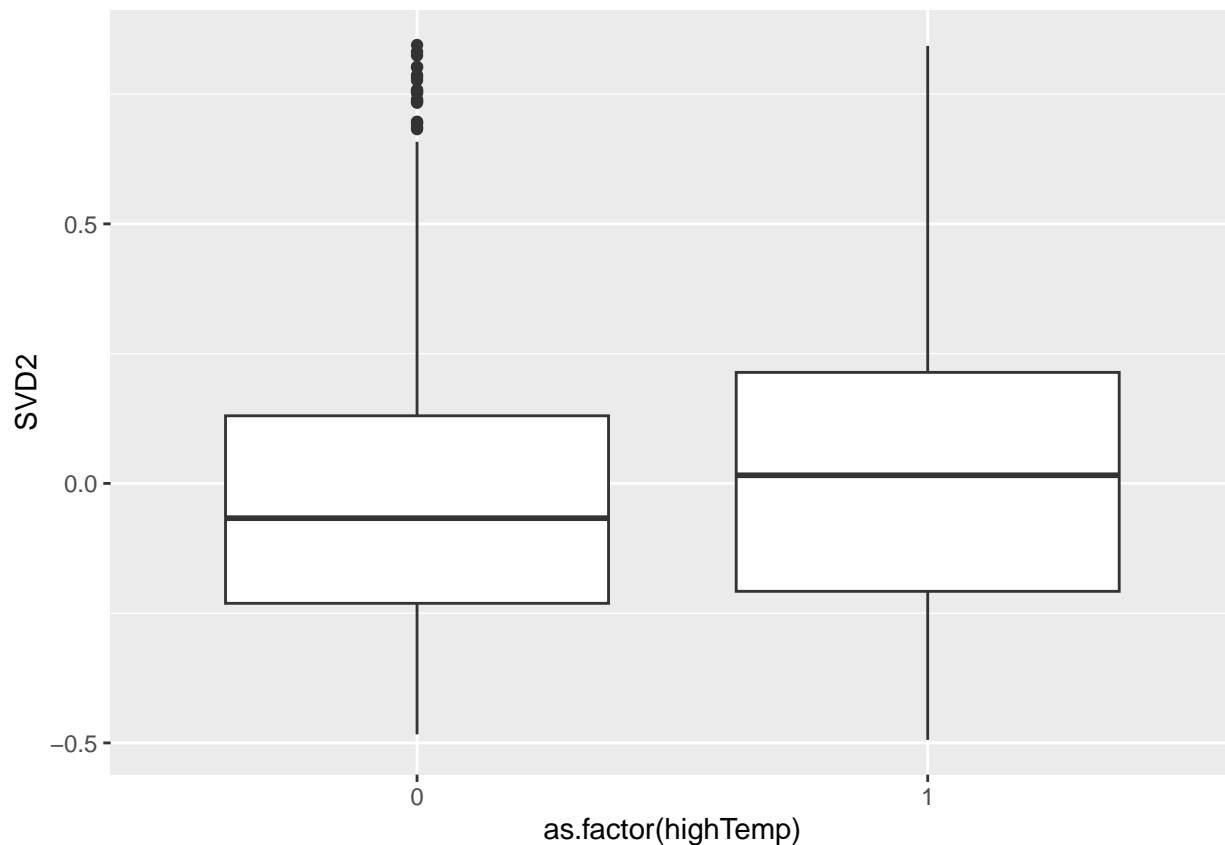


temperature

```
ggplot(reg_data, aes(x = as.factor(highTemp), y = SVD1)) + geom_boxplot()
```



```
ggplot(reg_data, aes(x = as.factor(highTemp), y = SVD2)) + geom_boxplot()
```



Clustering of observations

```

ICCest(worker_id,SVD1,reg_data) #CI does not contain zero; significant
#> Warning in ICCest(worker_id, SVD1, reg_data): 'x' has been coerced to a factor
#> $ICC
#> [1] 0.4679765
#>
#> $LowerCI
#> [1] 0.3809611
#>
#> $UpperCI
#> [1] 0.5714229
#>
#> $N
#> [1] 61
#>
#> $k
#> [1] 23.02129
#>
#> $varw
#> [1] 0.1109908
#>
#> $vara
#> [1] 0.09762927
ICCest(worker_id,SVD2,reg_data) #CI does not contain zero; significant
#> Warning in ICCest(worker_id, SVD2, reg_data): 'x' has been coerced to a factor

```

```

#> $ICC
#> [1] 0.1101448
#>
#> $LowerCI
#> [1] 0.07055893
#>
#> $UpperCI
#> [1] 0.1707893
#>
#> $N
#> [1] 61
#>
#> $k
#> [1] 23.02129
#>
#> $varw
#> [1] 0.07647555
#>
#> $vara
#> [1] 0.009466017

#suggests multilevel models are appropriate for these data

ICCest(prompt_code,SVD1,reg_data) #CI does contains zero; not significant
#> Warning in ICCest(prompt_code, SVD1, reg_data): 'x' has been coerced to a
#> factor
#> $ICC
#> [1] -0.006923188
#>
#> $LowerCI
#> [1] -0.009977924
#>
#> $UpperCI
#> [1] 0.001177918
#>
#> $N
#> [1] 20
#>
#> $k
#> [1] 71.70869
#>
#> $varw
#> [1] 0.2063118
#>
#> $vara
#> [1] -0.001418515
ICCest(prompt_code,SVD2,reg_data) #CI contains zero; not significant
#> Warning in ICCest(prompt_code, SVD2, reg_data): 'x' has been coerced to a
#> factor
#> $ICC
#> [1] 0.007406709
#>
#> $LowerCI

```

```

#> [1] -0.001636156
#>
#> $UpperCI
#> [1] 0.03086913
#>
#> $N
#> [1] 20
#>
#> $k
#> [1] 71.70869
#>
#> $varw
#> [1] 0.08498509
#>
#> $vara
#> [1] 0.0006341568

```

Regression analysis

SVD1

```

# mod.x.1 = lmerTest::lmer(SVD1 ~ genre*ownershipMetadata*highTemp + (1|worker_id),data = reg_data)
#
# mod.x.2 = lmerTest::lmer(SVD1 ~ genre*ownershipMetadata + genre*highTemp + ownershipMetadata*highTemp
#
#
# anova(mod.x.1,mod.x.2)
#
# mod.x.3 = lmerTest::lmer(SVD1 ~ genre + ownershipMetadata + highTemp + (1|worker_id),data = reg_data)
#
# anova(mod.x.2,mod.x.3)

mod.x.4 = lmerTest::lmer(SVD1 ~ ownershipMetadata + (1|worker_id),data = reg_data)

#anova(mod.x.3,mod.x.4)

summary(mod.x.4)
#> Linear mixed model fit by REML. t-tests use Satterthwaite's method [
#> lmerModLmerTest]
#> Formula: SVD1 ~ ownershipMetadata + (1 | worker_id)
#> Data: reg_data
#>
#> REML criterion at convergence: 1061.9
#>
#> Scaled residuals:
#>      Min       1Q   Median       3Q      Max
#> -3.0915 -0.6637 -0.0028  0.6290  3.1789
#>
#> Random effects:
#> Groups Name Variance Std.Dev.
#> worker_id (Intercept) 0.09539 0.3089
#> Residual 0.11026 0.3321
#> Number of obs: 1436, groups: worker_id, 61

```

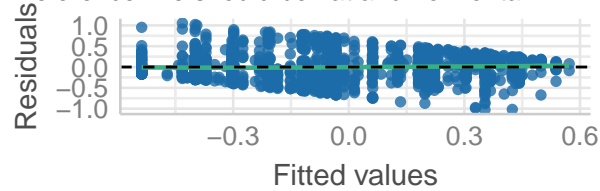
```
#>
#> Fixed effects:
#>               Estimate Std. Error      df t value Pr(>|t|)
#> (Intercept)      3.360e-03  4.593e-02 6.577e+01   0.073 0.941916
#> ownershipMetadatauser 9.557e-02  2.450e-02 1.434e+03  3.900 0.000101 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Correlation of Fixed Effects:
#>              (Intr)
#> ownrshpMtdt -0.317
```

```
check.x = check_model(mod.x.4, check = c("qq", "normality", "linearity", "homogeneity", "outliers", "reqq"))
#> Not enough model terms in the conditional part of the model to check for
#> multicollinearity.
```

```
check.x
```

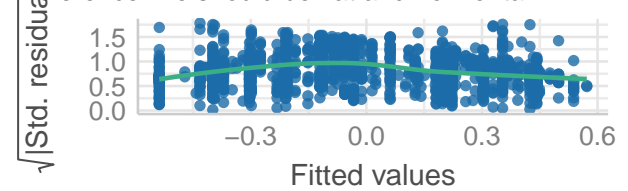
Linearity

Reference line should be flat and horizontal



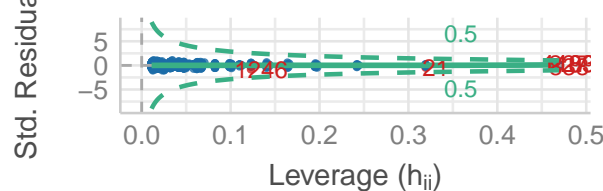
Homogeneity of Variance

Reference line should be flat and horizontal



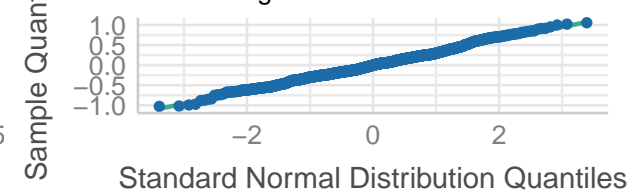
Influential Observations

Points should be inside the contour lines



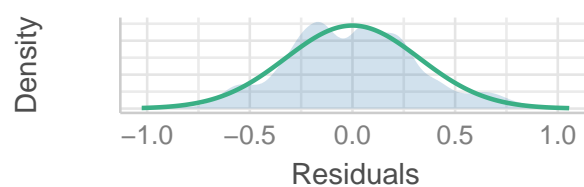
Normality of Residuals

Dots should fall along the line



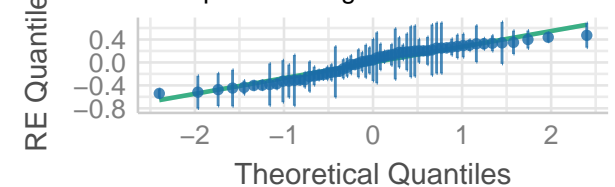
Normality of Residuals

Distribution should be close to the normal curve



Normality of Random Effects (worker_id)

Dots should be plotted along the line



Check model

```
check_outliers(mod.x.4, "mahalanobis")
#> Converting missing values ('NA') into regular values currently not
#> possible for variables of class 'NULL'.
#> OK: No outliers detected.
#> - Based on the following method and threshold: mahalanobis (10.828).
#> - For variable: (Whole model)
```

```
cohensd = function(diff_,g1,g2){
  diff_/((sqrt((sd(g1)^2 + sd(g2)^2)/2))
}
```

```
diff_ = coefficients(mod.x.4)$worker_id$ownershipMetadata$user[1]
g1 = as.matrix(set$points$ownershipMetadata$user)[,"SVD1"]
g2 = as.matrix(set$points$ownershipMetadata$api)[,"SVD1"]

cohensd(diff_ = diff_,g1 = g1, g2 = g2)
#> [1] 0.2157699
```

Effect size

SVD2

```
#mod.y.1 = lmerTest::lmer(SVD2 ~ genre*ownershipMetadata*highTemp + (1|worker_id),data = reg_data)

mod.y.2 = lmerTest::lmer(SVD2 ~ genre*ownershipMetadata + genre*highTemp + ownershipMetadata*highTemp +

#anova(mod.y.1,mod.y.2)

#mod.y.3 = lmerTest::lmer(SVD2 ~ genre*highTemp + (1|worker_id),data = reg_data)

#anova(mod.y.2,mod.y.3) #prefer mod.y.2

summary(mod.y.2)
#> Linear mixed model fit by REML. t-tests use Satterthwaite's method [
#> lmerModLmerTest]
#> Formula:
#> SVD2 ~ genre * ownershipMetadata + genre * highTemp + ownershipMetadata *
#>      highTemp + (1 | worker_id)
#> Data: reg_data
#>
#> REML criterion at convergence: 435.2
#>
#> Scaled residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.2982 -0.6969 -0.1193  0.4837  3.4913
#>
#> Random effects:
#> Groups      Name                Variance Std.Dev.
#> worker_id (Intercept) 0.01008  0.1004
#> Residual              0.07392  0.2719
#> Number of obs: 1436, groups: worker_id, 61
#>
#> Fixed effects:
#>                                     Estimate Std. Error      df t value
#> (Intercept)                        0.01230    0.02997  232.67943    0.411
#> genrecreative                      -0.02239    0.02725  1419.38959   -0.822
#> ownershipMetadata$user             -0.10556    0.03052  1339.07994   -3.459
#> highTemp                          0.11331    0.02715  1400.36167    4.173
```



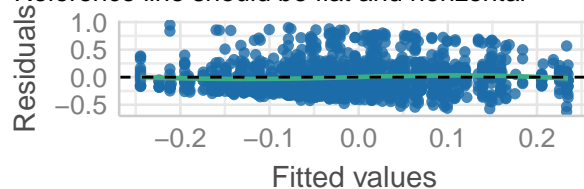
```
#> genrecreative:ownershipMetadatauser 0.05532 0.03049 1421.28379 1.815
#> genrecreative:highTemp -0.06380 0.02977 1396.09968 -2.143
#> ownershipMetadatauser:highTemp -0.03290 0.02946 1399.14816 -1.117
#> Pr(>|t|)
#> (Intercept) 0.681817
#> genrecreative 0.411405
#> ownershipMetadatauser 0.000559 ***
#> highTemp 3.18e-05 ***
#> genrecreative:ownershipMetadatauser 0.069809 .
#> genrecreative:highTemp 0.032306 *
#> ownershipMetadatauser:highTemp 0.264368
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Correlation of Fixed Effects:
#> (Intr) gnrcrt ownrsM hghTmP gnrc:M gnrc:T
#> genrecreativ -0.618
#> ownrshpMtdt -0.661 0.472
#> highTemp -0.587 0.466 0.459
#> gnrcrtv:wmM 0.385 -0.644 -0.607 -0.157
#> gnrcrtv:hgT 0.379 -0.620 -0.177 -0.653 0.156
#> ownrshpMt:T 0.303 -0.059 -0.483 -0.537 0.000 0.013
```

```
check.y = check_model(mod.y.2, check = c("qq", "normality", "linearity", "homogeneity", "outliers", "reqq"))
```

```
check.y
```

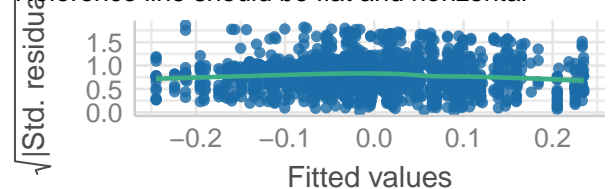
Linearity

Reference line should be flat and horizontal



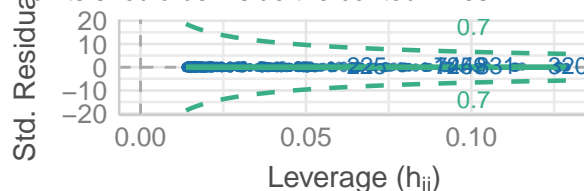
Homogeneity of Variance

Reference line should be flat and horizontal



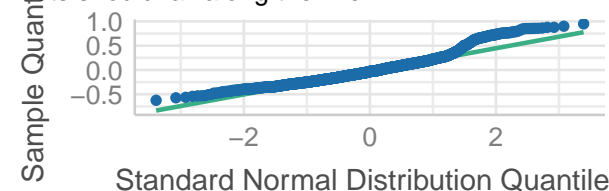
Influential Observations

Points should be inside the contour lines



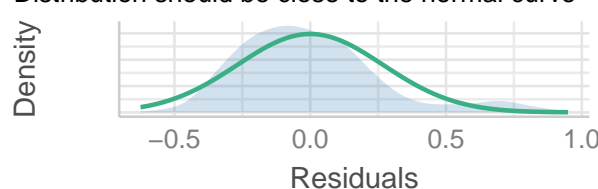
Normality of Residuals

Dots should fall along the line



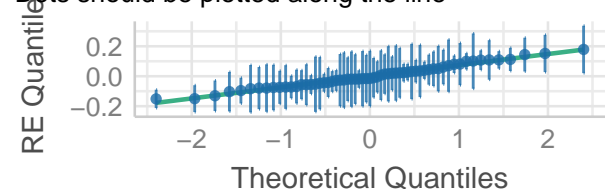
Normality of Residuals

Distribution should be close to the normal curve



Normality of Random Effects (worker_id)

Dots should be plotted along the line



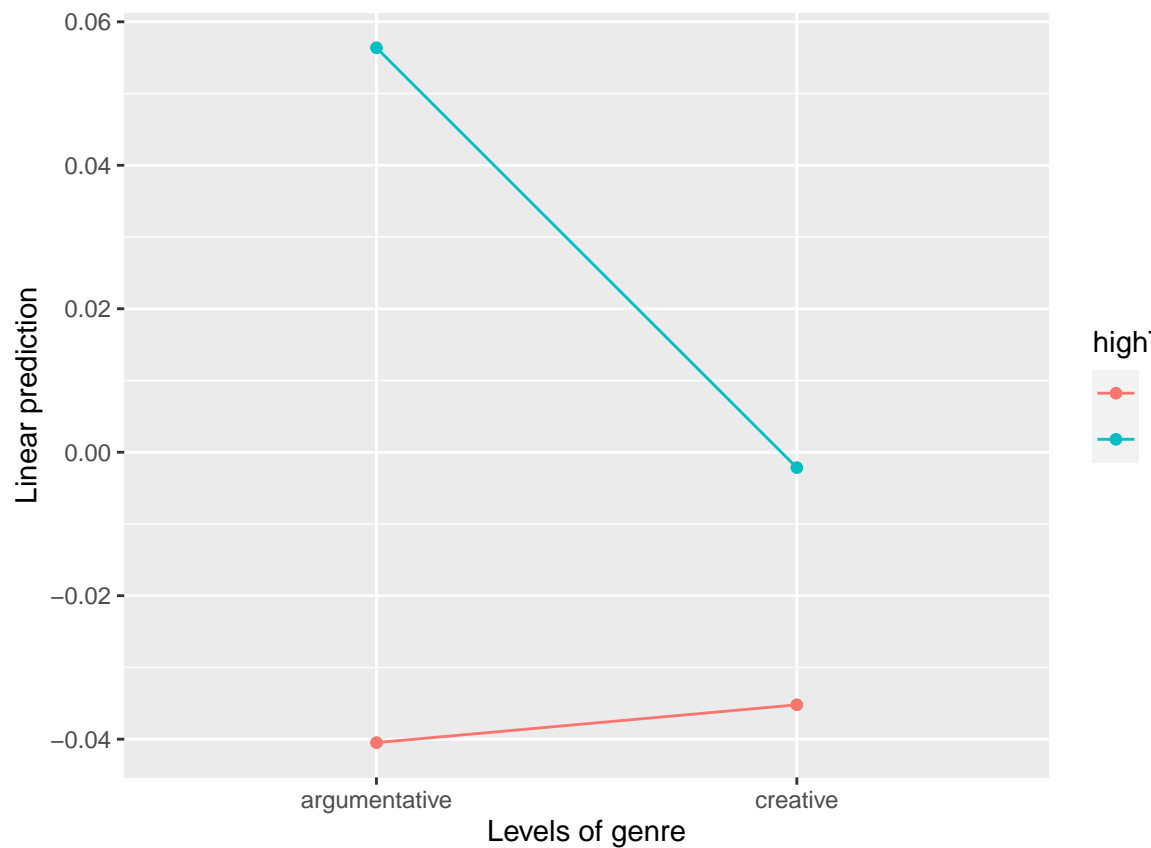
check model

```
check_outliers(mod.y.2,"mahalanobis")
#> Converting missing values (`NA`) into regular values currently not
#> possible for variables of class `NULL`.
#> OK: No outliers detected.
#> - Based on the following method and threshold: mahalanobis (13.816).
#> - For variable: (Whole model)
```

```
emm.y.1 = emmeans(mod.y.2, specs = pairwise ~ ownershipMetadata, weights = "proportional")
#> NOTE: Results may be misleading due to involvement in interactions
emm.y.2 = emmeans(mod.y.2, specs = pairwise ~ highTemp, weights = "proportional")
#> NOTE: Results may be misleading due to involvement in interactions
```

Estimate marginal means

```
emmip(mod.y.2, highTemp ~ genre)
```



Viewing interactions

```
#x = ref_grid(mod.y.2)
#broom::tidy(x)

diff_ = 0.0899
g1 = as.matrix(set$points$ownershipMetadata$user)[,"SVD2"]
g2 = as.matrix(set$points$ownershipMetadata$api)[,"SVD2"]
```

```
cohensd(diff_ = diff_, g1 = g1, g2 = g2)
#> [1] 0.3113426
```

Effect sizes (ownership)

```
diff_ = -0.067
g1 = as.matrix(set$points$highTemp$'1')[, "SVD2"]
g2 = as.matrix(set$points$highTemp$'0')[, "SVD2"]

cohensd(diff_ = diff_, g1 = g1, g2 = g2)
#> [1] -0.2303132
```

Effect sizes (temp)

ENA plots

Mean networks (ownership)

```
user_pts = as.matrix(set$points$ownershipMetadata$user)
api_pts = as.matrix(set$points$ownershipMetadata$api)

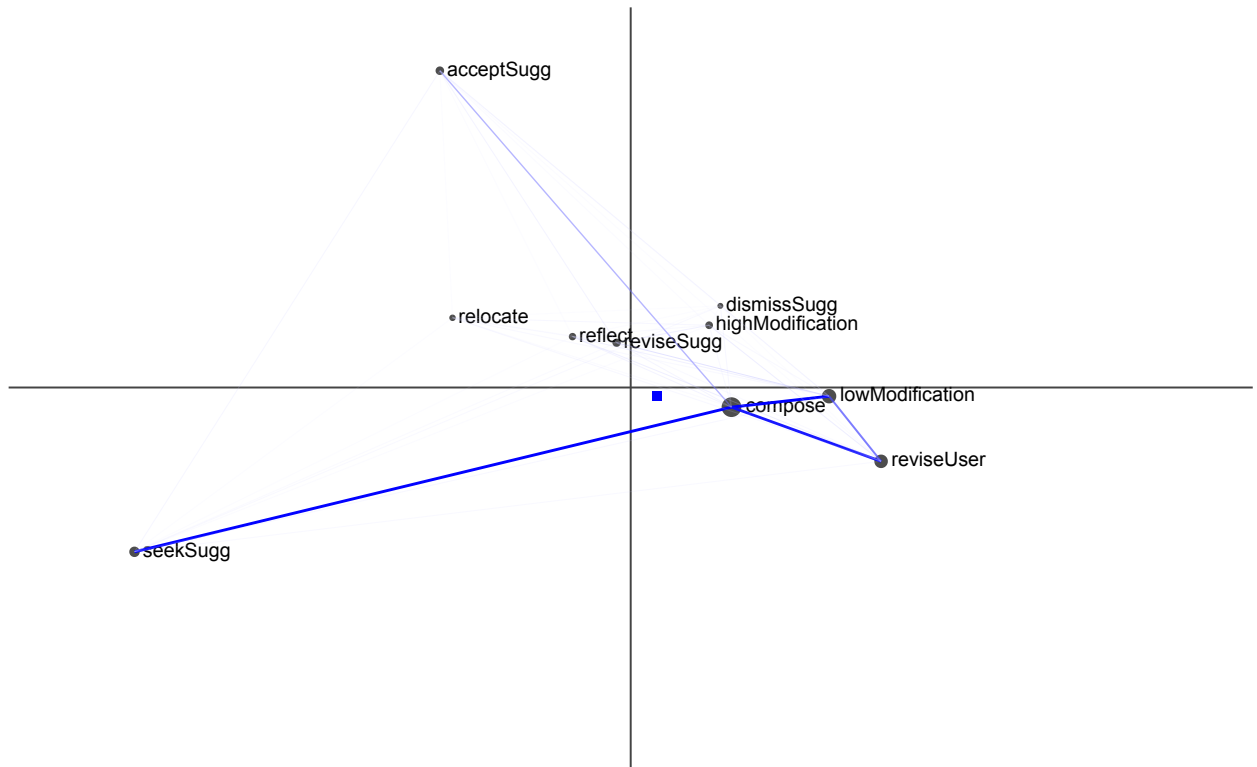
user_net = colMeans(as.matrix(set$line.weights$ownershipMetadata$user))
api_net = colMeans(as.matrix(set$line.weights$ownershipMetadata$api))

plot_user = ena.plot(set, scale.to = "network", title = "User") %>%
  # ena.plot.points(points = creat_pts, colors = c("blue")) %>%
  ena.plot.group(point = user_pts,
                 colors = c("blue"), confidence.interval = "none") %>%
  ena.plot.network(network = user_net, colors = c("blue") )

plot_api = ena.plot(set, scale.to = "network", title = "Api") %>%
  # ena.plot.points(points = arg_pts, colors = c("red")) %>%
  ena.plot.group(point = api_pts,
                 colors = c("red"), confidence.interval = "none") %>%
  ena.plot.network(network = api_net, colors = c("red") )

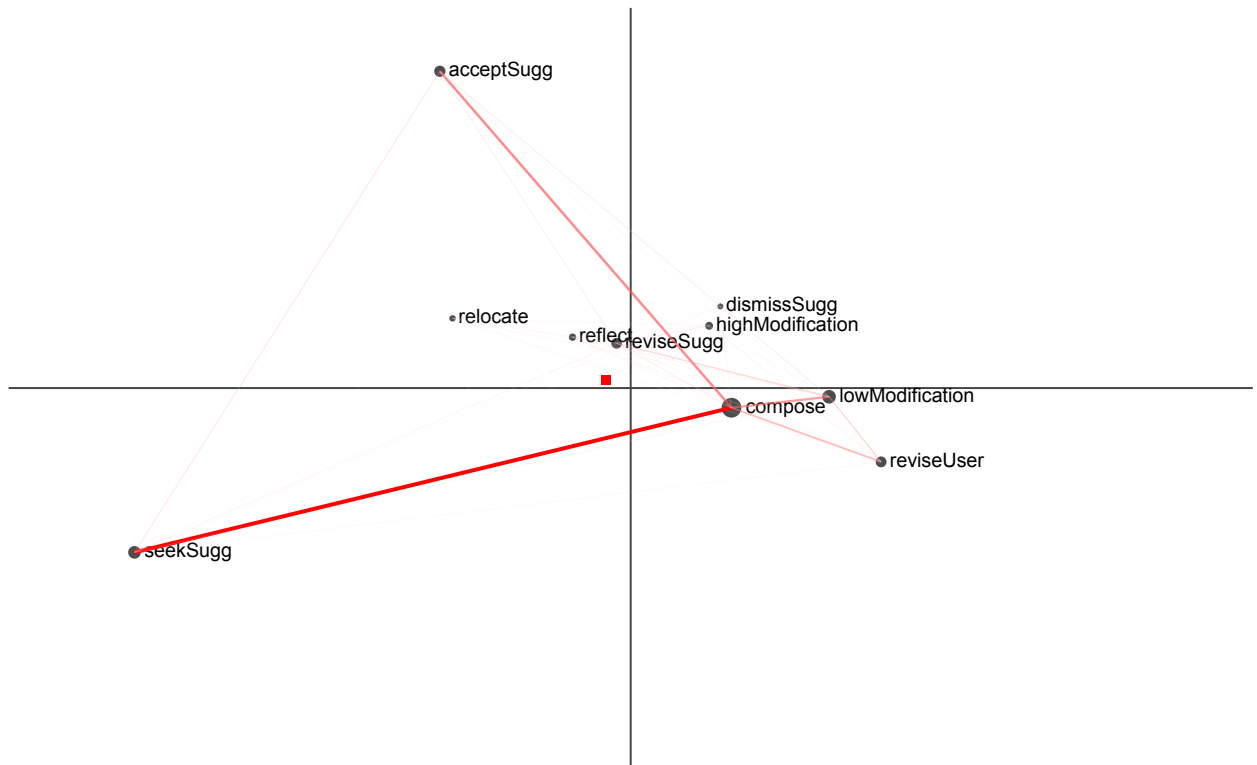
plot_user$plot
```

User



plot_api\$plot

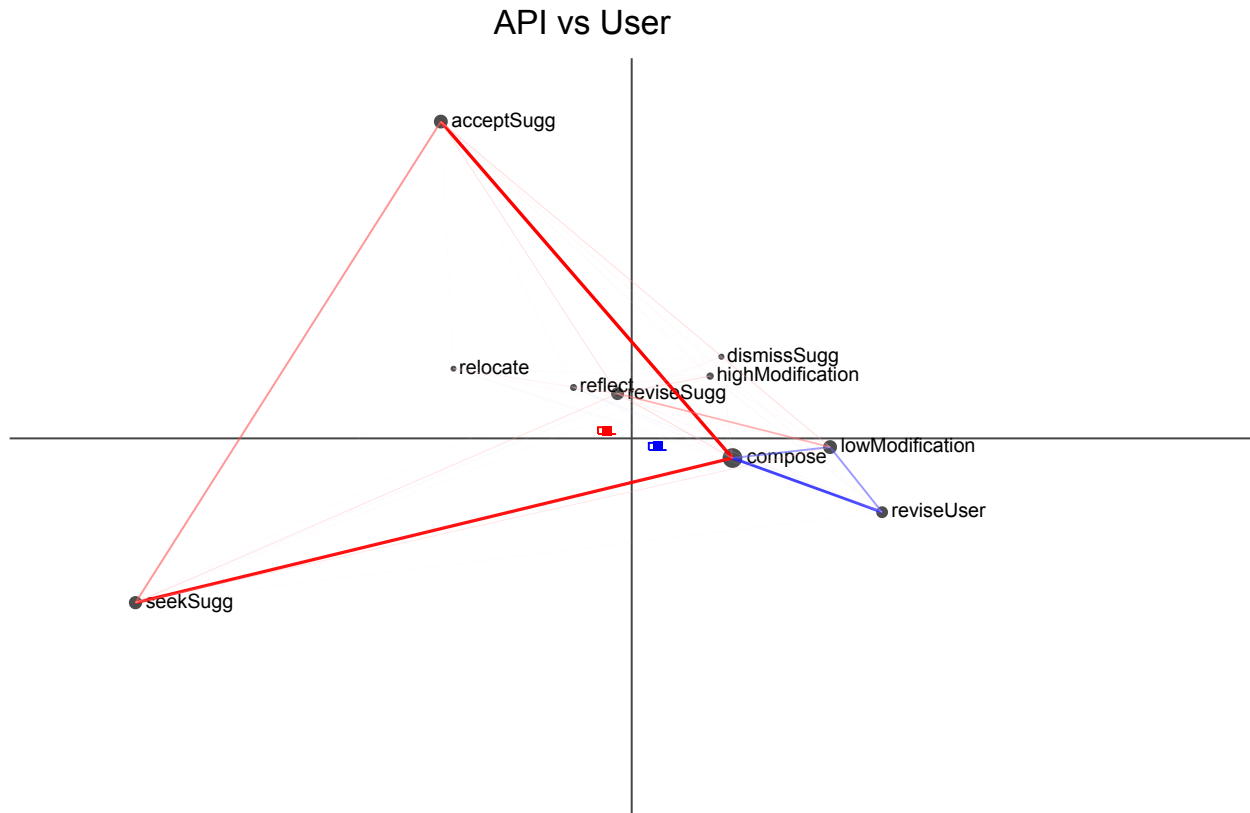
Api



```
## Network subtraction (ownership)
```

```
net_mult = 3
```

```
plot_sub = ena.plot(set, scale.to = "network", title = "API vs User") %>%
  ena.plot.group(point = user_pts,
    colors = c("blue"), confidence.interval = "box") %>%
  ena.plot.group(point = api_pts,
    colors = c("red"), confidence.interval = "box") %>%
  ena.plot.network(network = (user_net - api_net) * net_mult, colors = c("blue", "red"))
plot_sub$plot
```



```
## Mean networks (temp)
```

```
high_pts = as.matrix(set$points$highTemp$'1')
```

```
low_pts = as.matrix(set$points$highTemp$'0')
```

```
high_net = colMeans(as.matrix(set$line.weights$highTemp$'1'))
```

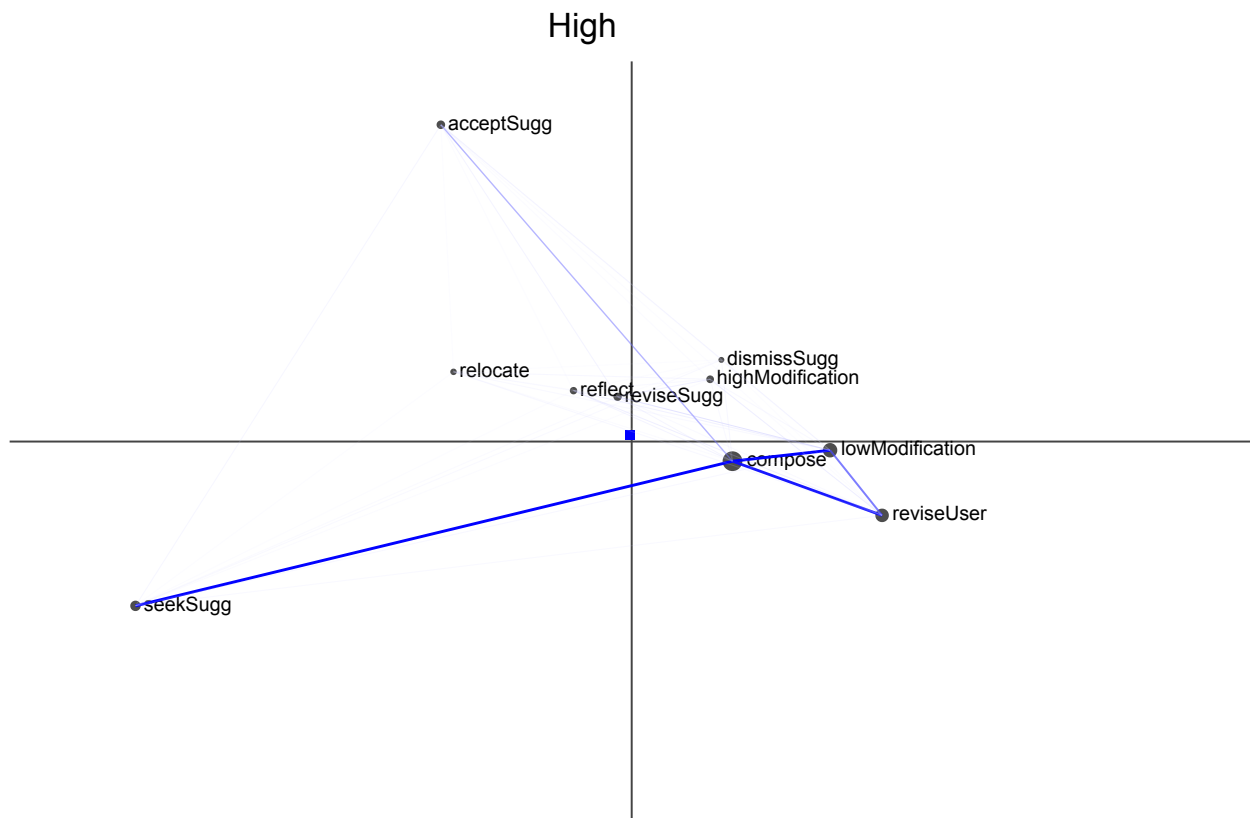
```
low_net = colMeans(as.matrix(set$line.weights$highTemp$'0'))
```

```
plot_high = ena.plot(set, scale.to = "network", title = "High") %>%
  # ena.plot.points(points = creat_pts, colors = c("blue")) %>%
  ena.plot.group(point = high_pts,
    colors = c("blue"), confidence.interval = "none") %>%
  ena.plot.network(network = user_net, colors = c("blue"))
```

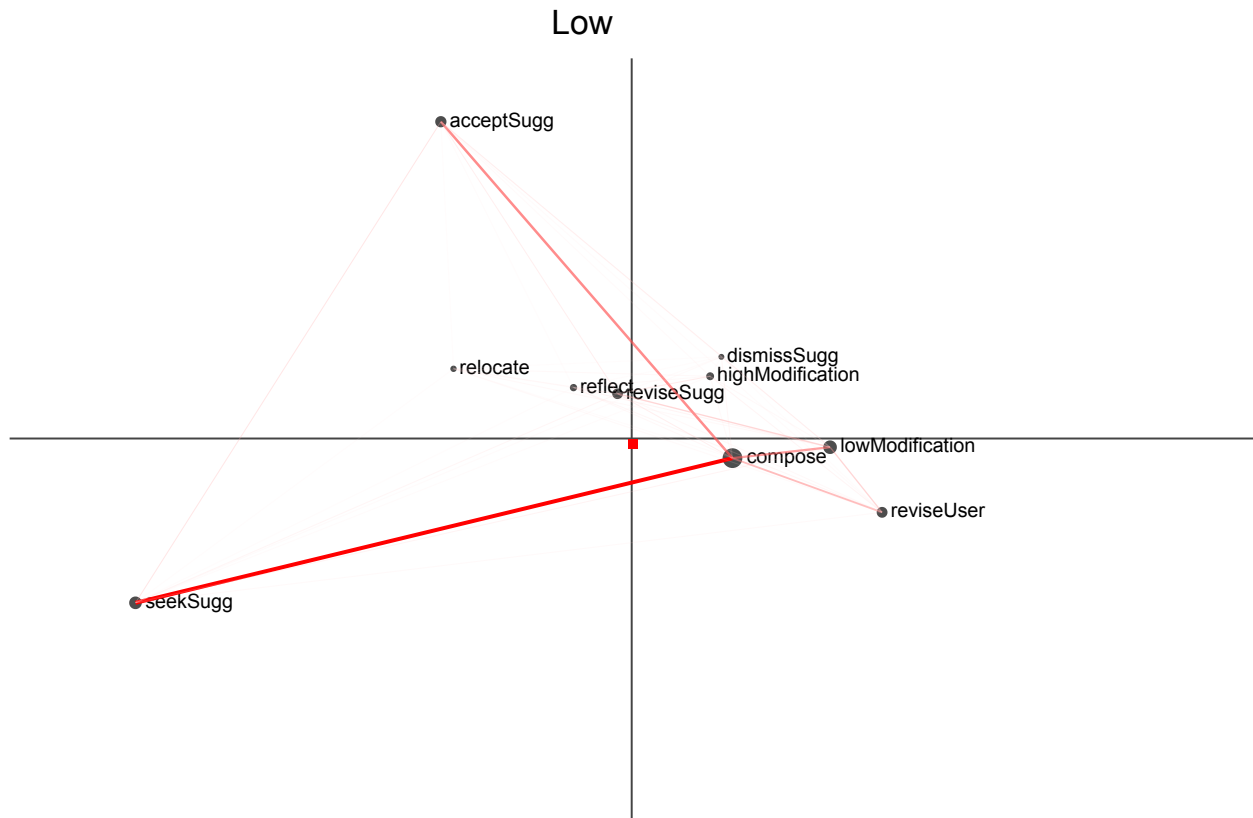
```
plot_low = ena.plot(set, scale.to = "network", title = "Low") %>%
  # ena.plot.points(points = arg_pts, colors = c("red")) %>%
```

```
ena.plot.group(point = low_pts,
               colors = c("red"), confidence.interval = "none") %>%
ena.plot.network(network = api_net, colors = c("red") )
```

```
plot_high$plot
```



```
plot_low$plot
```



Network subtraction (temp)

```
net_mult = 3

plot_sub_temp = ena.plot(set, scale.to = "network", title = "High vs Low") %>%
  ena.plot.group(point = high_pts,
                 colors = c("blue"), confidence.interval = "box") %>%
  ena.plot.group(point = low_pts,
                 colors = c("red"), confidence.interval = "box") %>%
  ena.plot.network(network = (high_net - low_net) * net_mult, colors = c("blue", "red") )
plot_sub_temp$plot
```

High vs Low

