coAuthor-ENA

Zach

2023-08-31

Contents

Load packages

```r
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_new_v2.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',
  "cursorFwd",
  "cursorBwd",
  "cursorSelect",
  #"reopenSugg",
  "hoverSugg"
)
```

```
codes = data1[,codeCols]

#mask =

meta = data1[,c("genre",
                "highTemp",
                "ownershipMetadata",
                "prompt_code"
                )]
```

Run ENA accumulation

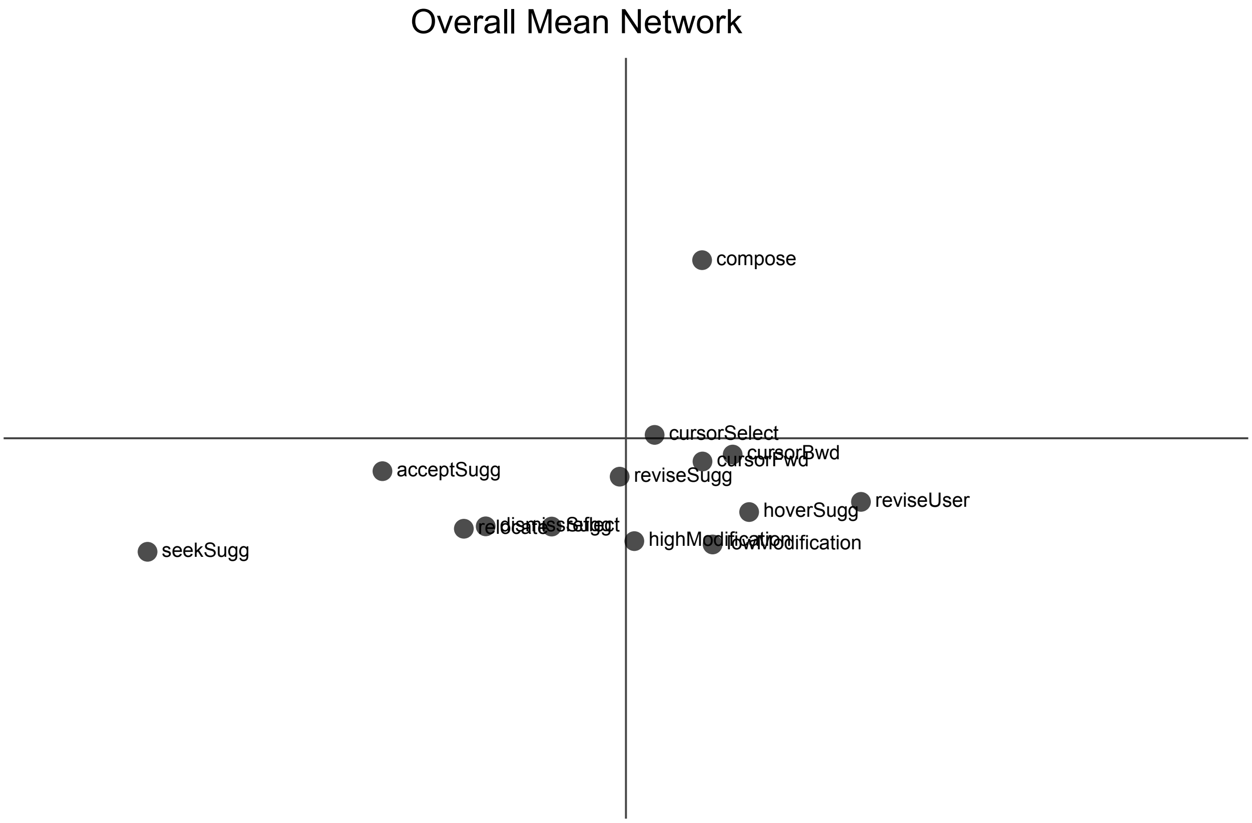
```
accum =
  ena.accumulate.data(
  units = units,
  conversation = conversation,
  codes = codes,
  metadata = meta,
  #mask = mask,
  window.size.back = "inf" # each line in the conversation can connect back to the first line--allows f
)
```

Run ENA dimensional reduction

```
set = ena.make.set(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
```



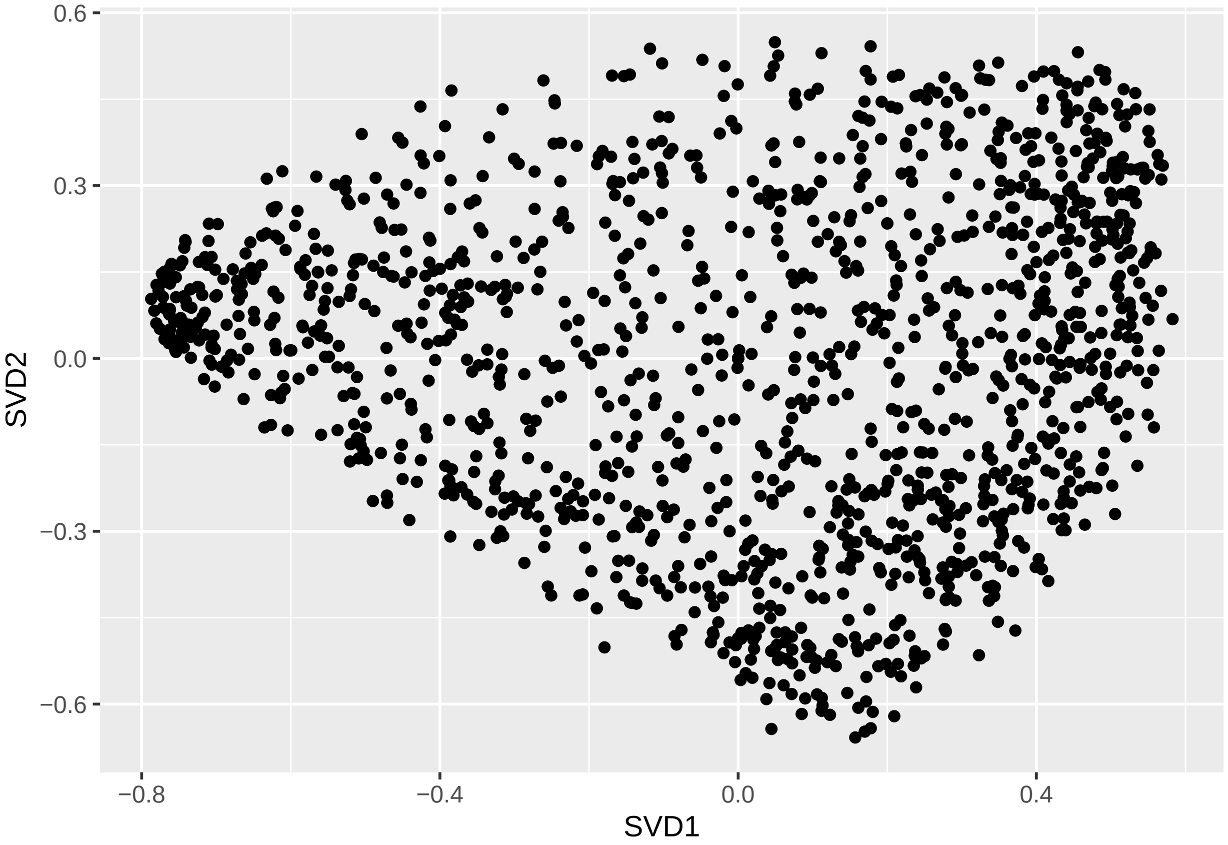
Statistical tests

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

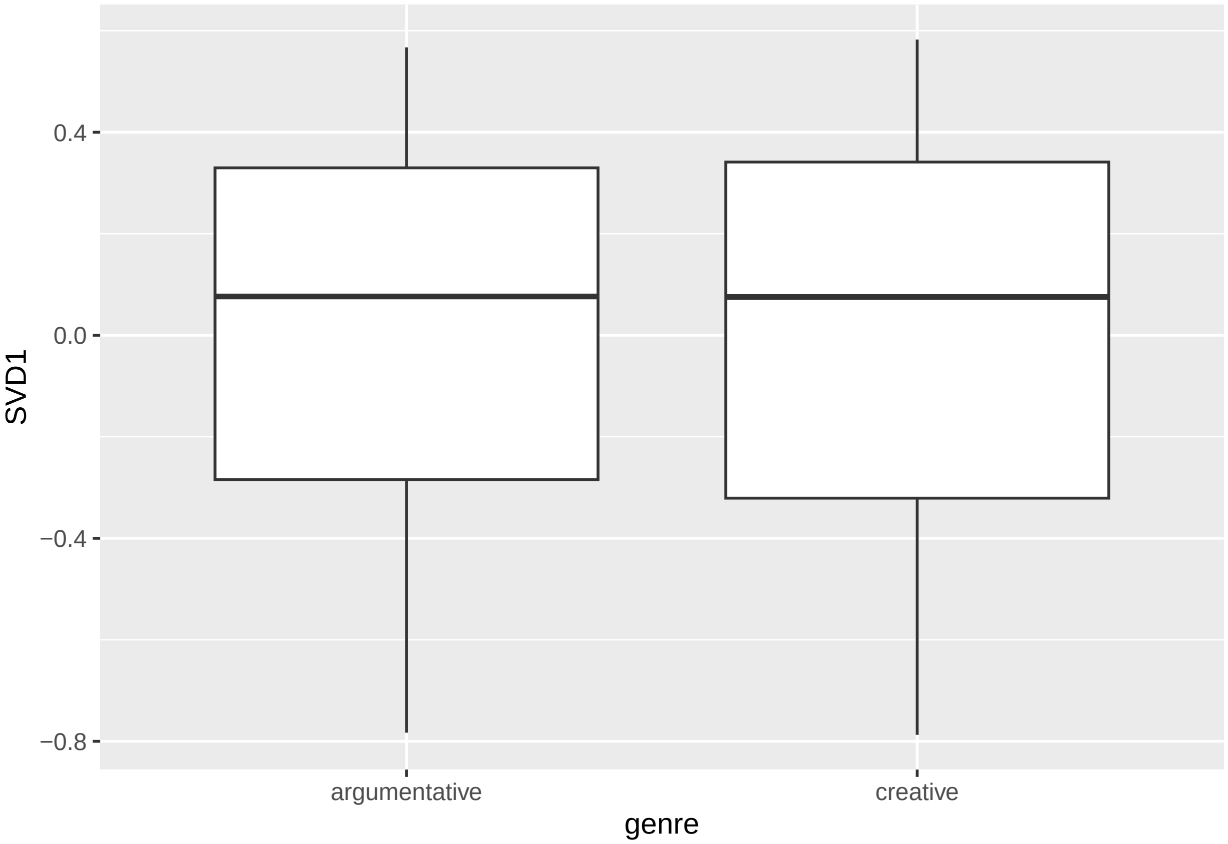
```
ggplot(reg_data,aes(x = SVD1, y = SVD2)) + geom_point()
```



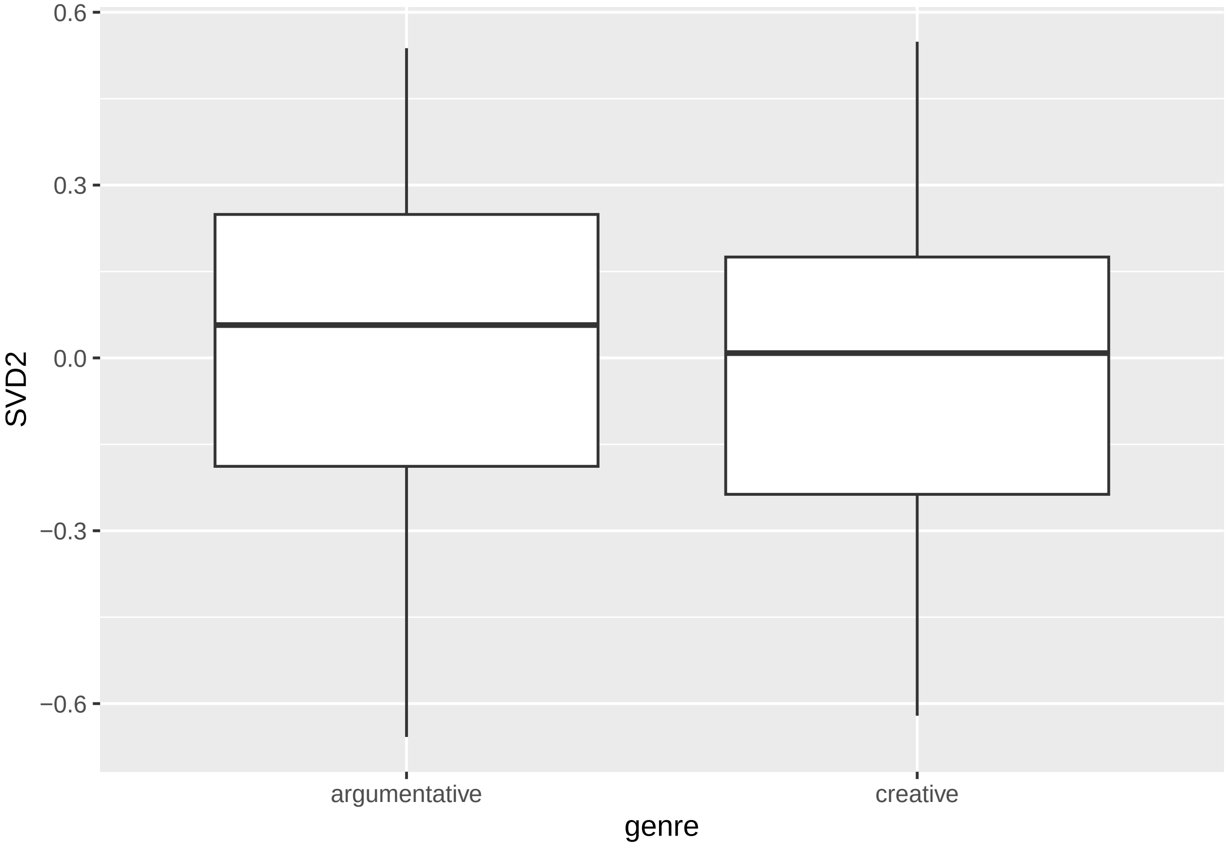
Checking other groups

genre

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

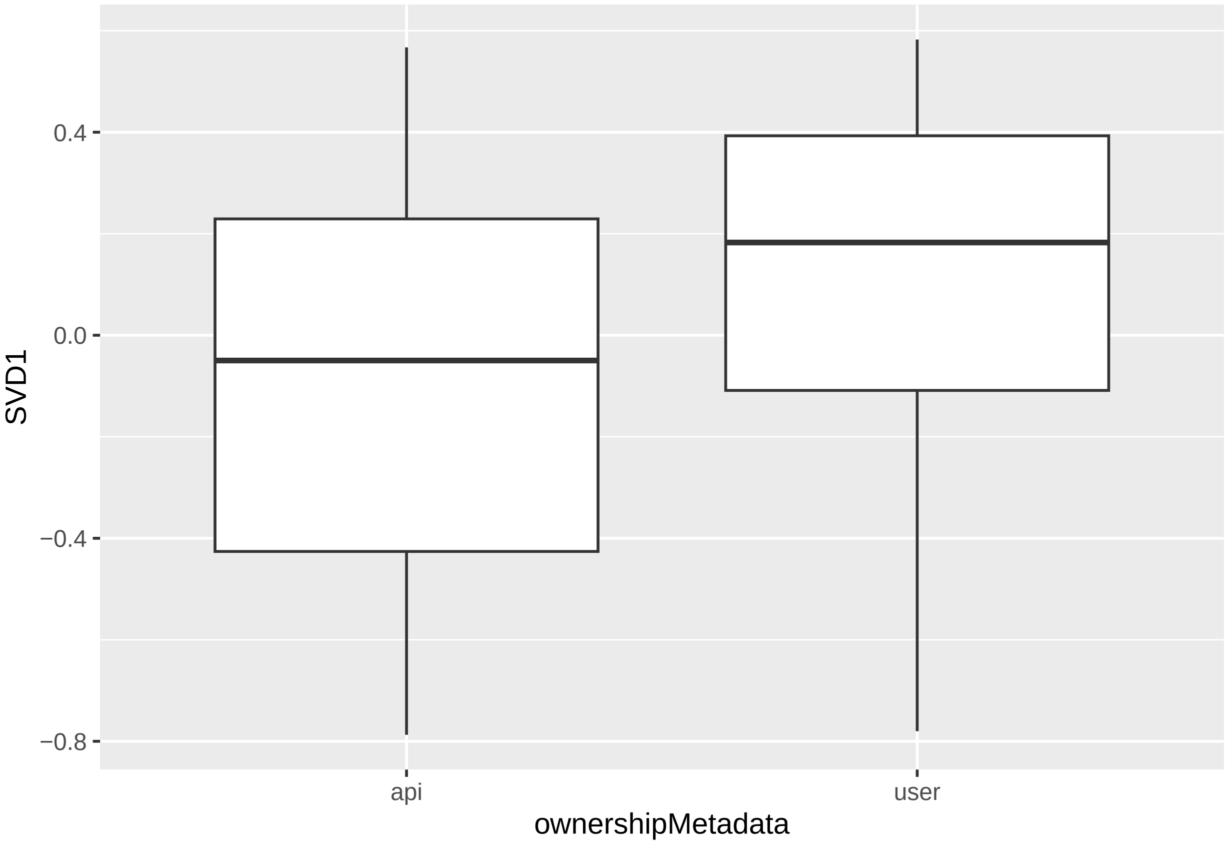


```r
ggplot(reg_data, aes(x = genre, y = SVD2)) + geom_boxplot()
```

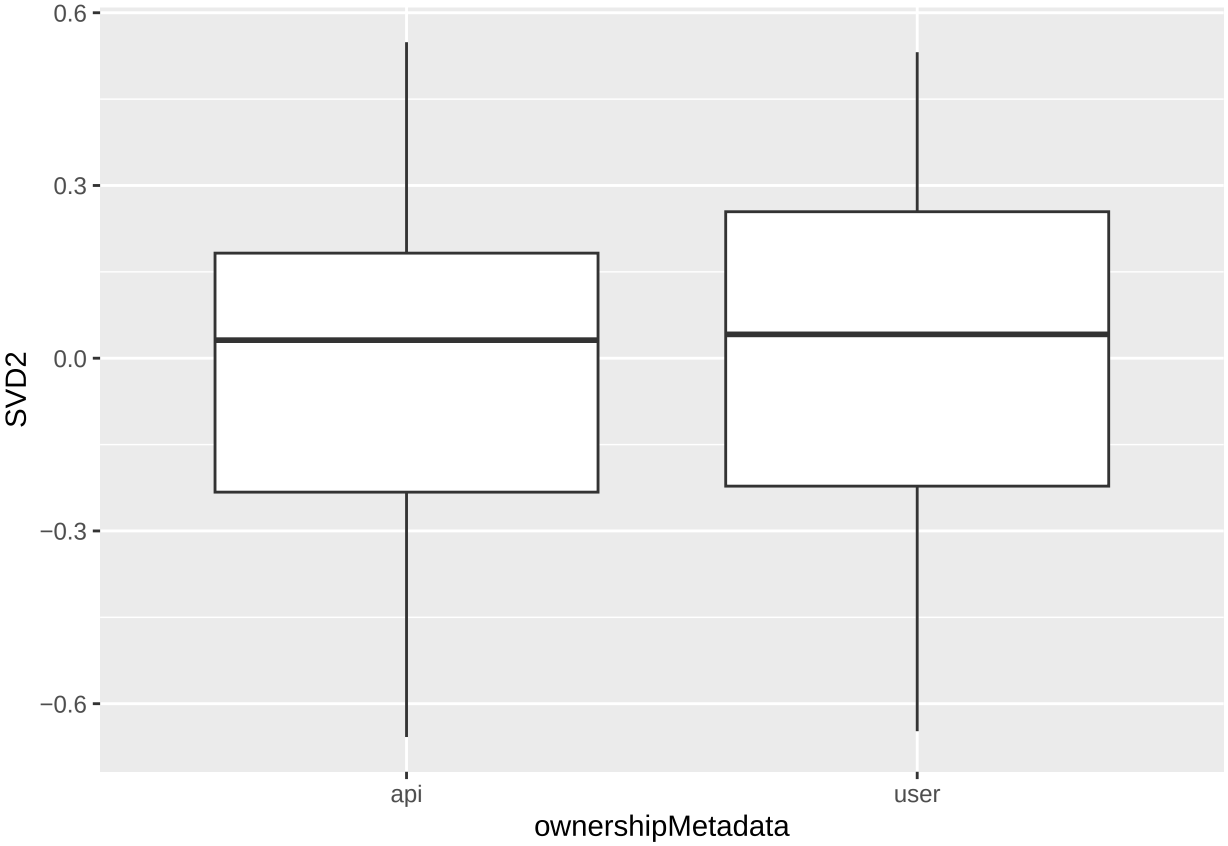


### ownership

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

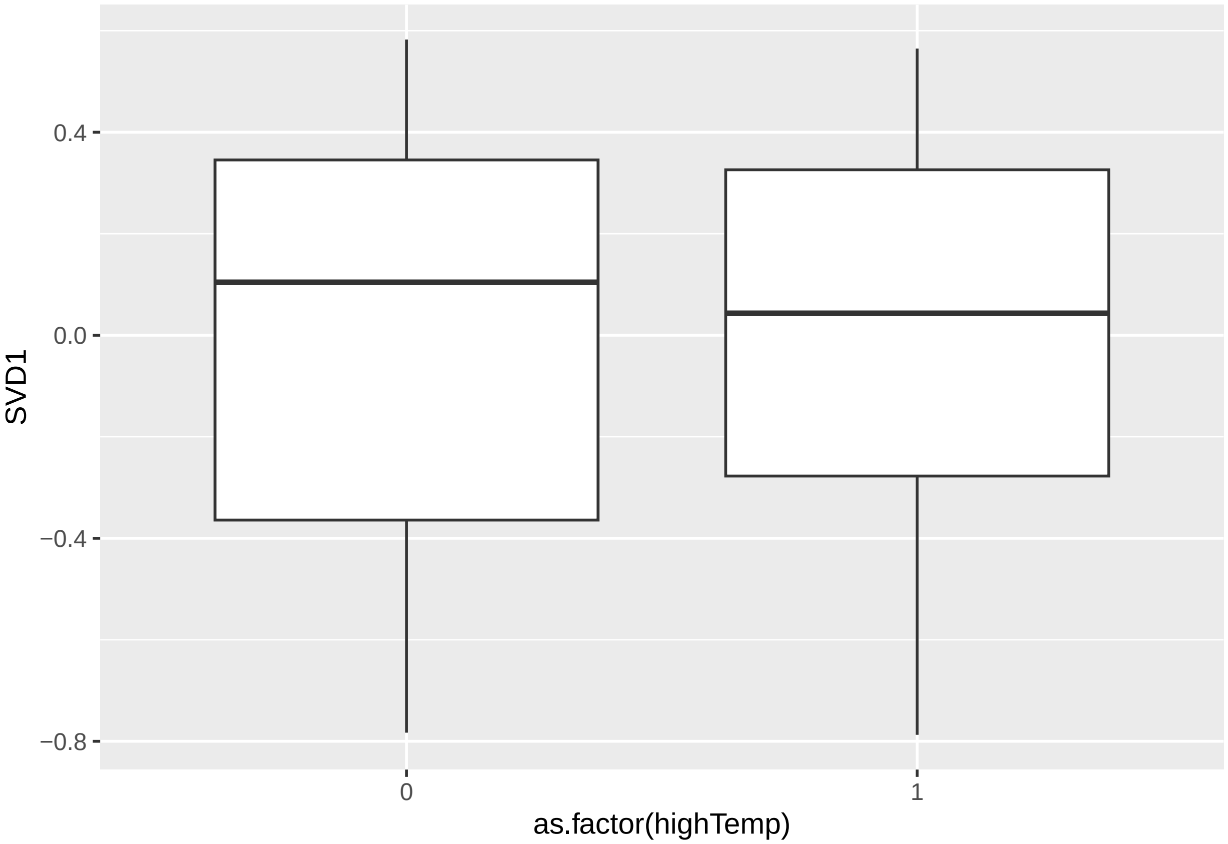


```r
ggplot(reg_data, aes(x = ownershipMetadata, y = SVD2)) + geom_boxplot()
```

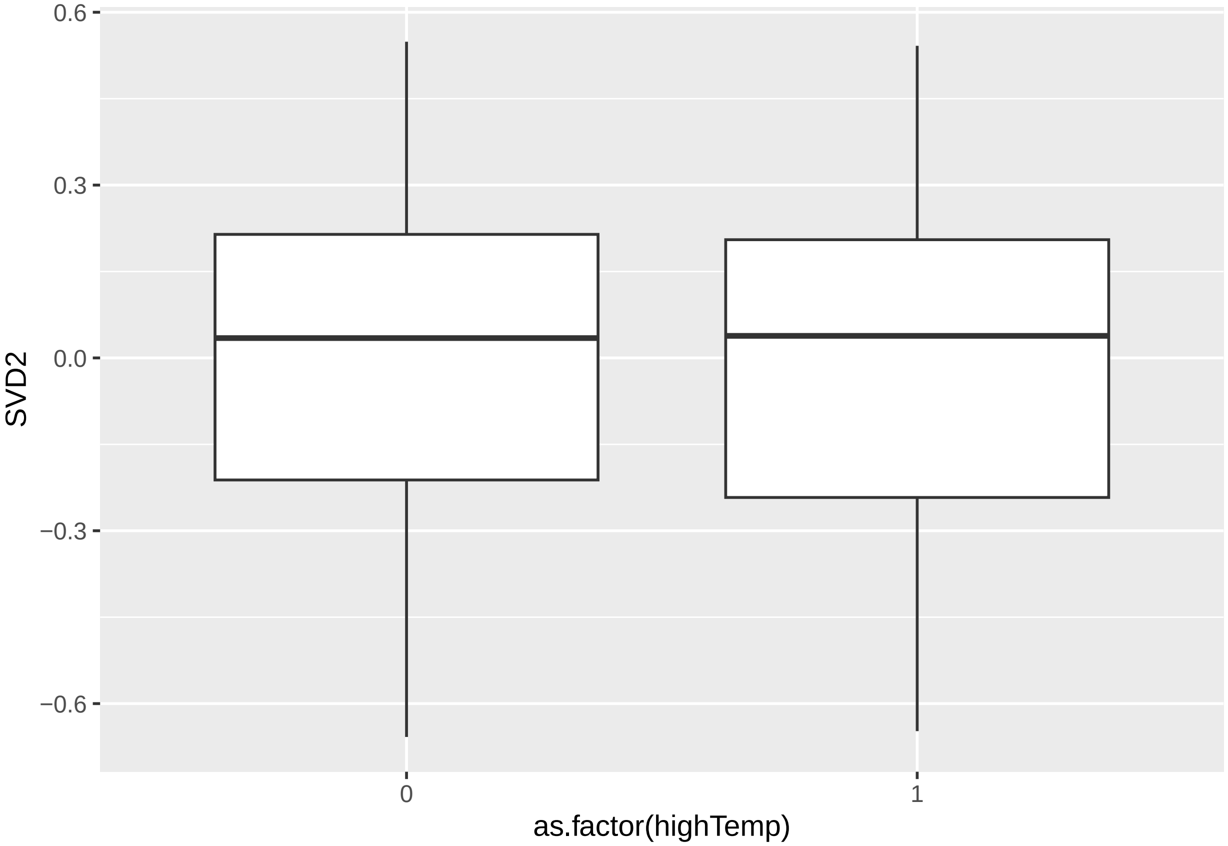


temperature

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



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



Clustering of observations

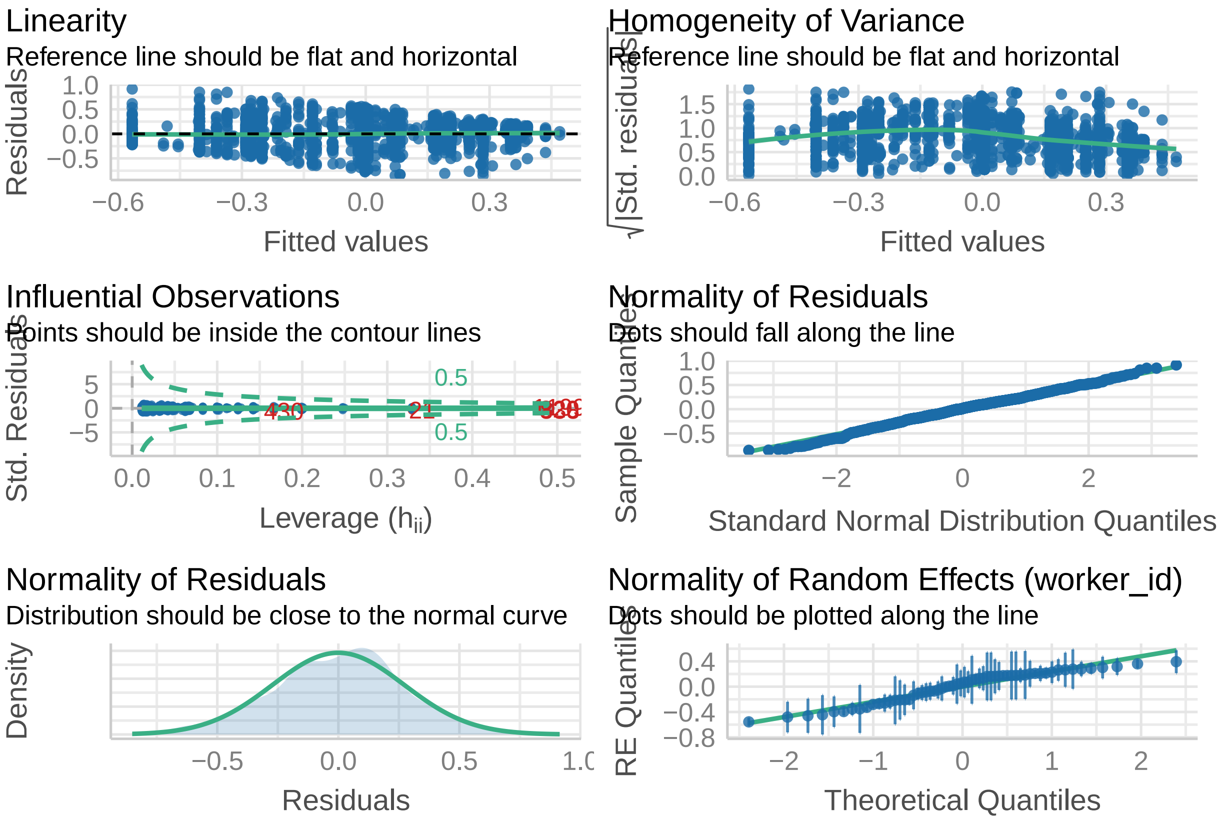
```
ICCest(worker_id,SVD1,reg_data) #CI does not contain zero; significant
#> Warning in ICCest(worker_id, SVD1, reg_data): 'x' has been coerced to a factor
#> $ICC
#> [1] 0.4986186
#>
#> $LowerCI
#> [1] 0.4102973
#>
#> $UpperCI
#> [1] 0.6015717
#>
#> $N
#> [1] 60
#>
#> $k
#> [1] 23.39389
#>
#> $varw
#> [1] 0.0774089
#>
#> $vara
#> [1] 0.07698236
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] 0.2856144
#>
#> $LowerCI
#> [1] 0.2140536
#>
#> $UpperCI
#> [1] 0.3825997
#>
#> $N
#> [1] 60
#>
#> $k
#> [1] 23.39389
#>
#> $varw
#> [1] 0.05525751
#>
#> $vara
#> [1] 0.02209219

#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] -0.004754699
#>
#> $LowerCI
#> [1] -0.008726362
#>
#> $UpperCI
#> [1] 0.005742934
#>
#> $N
#> [1] 20
#>
#> $k
#> [1] 71.65773
#>
#> $varw
#> [1] 0.152189
#>
#> $vara
#> [1] -0.0007201885
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] 0.007706712
#>
#> $LowerCI
#> [1] -0.001464712
#> 
#> $UpperCI
#> [1] 0.03149145
#> 
#> $N
#> [1] 20
#> 
#> $k
#> [1] 71.65773
#> 
#> $varw
#> [1] 0.07596133
#> 
#> $vara
#> [1] 0.0005899588
```

Regression analysis

SVD1

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

#confint(mod.x.1)

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

#confint(mod.x.2)

#suggests only ownership is signficant

mod.x.3 = lmerTest::lmer(SVD1 ~ ownershipMetadata + (1|worker_id),data = reg_data)

summary(mod.x.3)
#> 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: 547.7
#> 
#> Scaled residuals: 
#>     Min      1Q  Median      3Q     Max 
#> -3.0717 -0.6265  0.0493  0.6339  3.2976 
#> 
#> Random effects:
#>  Groups    Name        Variance Std.Dev.
#>  worker_id (Intercept) 0.07246  0.2692  
#>  Residual              0.07685  0.2772  
#> Number of obs: 1435, groups:  worker_id, 60
#> 
#> Fixed effects:
#>                          Estimate Std. Error         df t value Pr(>|t|)
#> (Intercept)            -9.616e-03  3.982e-02  6.303e+01  -0.241     0.81
#> ownershipMetadatauser   8.472e-02  2.049e-02  1.433e+03   4.134 3.77e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Correlation of Fixed Effects:
#>             (Intr)
#> ownrshpMtdt -0.304
```

```
check.x = check_model(mod.x.3, 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
```



Check model

```
check_outliers(mod.x.3,"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)
```

```
demean = function(x){
  return(x - mean(x))
}

cohensd = function(diff_,x1,x2){
  num = diff_
  denom = sqrt((sum(demean(x1)^2) + sum(demean(x2)^2))/(length(x1) + length(x2) - 2))
  return(num/denom)

}
```

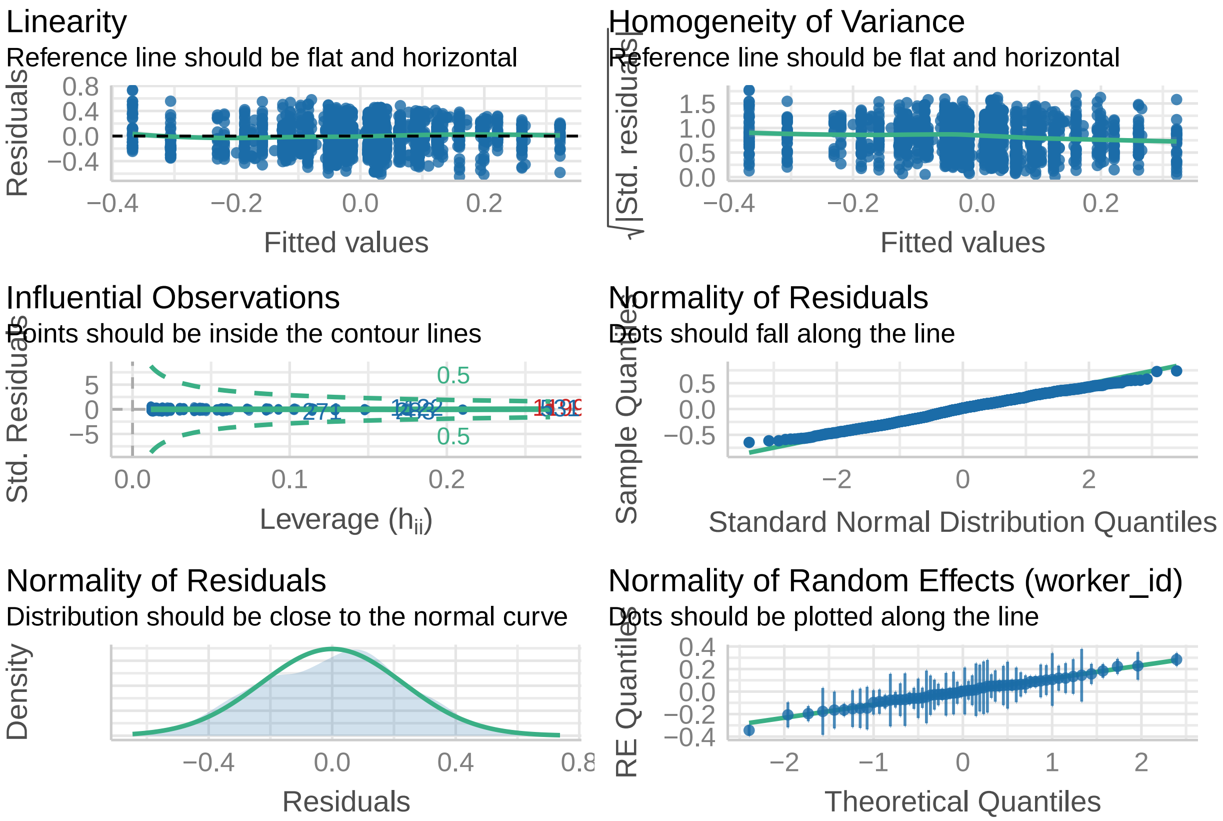
```
diff_ = coefficients(mod.x.3)$worker_id$ownershipMetadatauser[1]
x1 = as.matrix(set$points$ownershipMetadata$user)[,"SVD1"]
x2 = as.matrix(set$points$ownershipMetadata$api)[,"SVD1"]

cohensd(diff_ = diff_,x1,x2)
#> [1] 0.2246755
```

Effect size

SVD2

```
# mod.y.1 = lmerTest::lmer(SVD2 ~ genre*ownershipMetadata*highTemp + (1|worker_id),data = reg_data)
#
# summary(mod.y.1)
#
#
# mod.y.2 = lmerTest::lmer(SVD2 ~ genre + ownershipMetadata + highTemp  + (1|worker_id),data = reg_data
#
# summary(mod.y.2)

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

summary(mod.y.3)
#> Linear mixed model fit by REML. t-tests use Satterthwaite's method [
#> lmerModLmerTest]
#> Formula: SVD2 ~ genre + (1 | worker_id)
#>    Data: reg_data
#>
#> REML criterion at convergence: 9.3
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -2.7701 -0.7390  0.0760  0.6919  3.1611
#>
#> Random effects:
#>  Groups    Name        Variance Std.Dev.
#>  worker_id (Intercept) 0.01918  0.1385
#>  Residual              0.05446  0.2334
#> Number of obs: 1435, groups:  worker_id, 60
#>
#> Fixed effects:
#>                 Estimate Std. Error         df t value Pr(>|t|)
#> (Intercept)      0.03788    0.02280   70.97054   1.661    0.101
#> genrecreative   -0.06144    0.01306 1417.27777  -4.704  2.8e-06 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Correlation of Fixed Effects:
#>             (Intr)
#> genrecreatv -0.372
```

```
check.y = check_model(mod.y.3, check = c("qq","normality","linearity", "homogeneity", "outliers","reqq"
#> Not enough model terms in the conditional part of the model to check for
#>   multicollinearity.

check.y
```



check model

```
check_outliers(mod.y.3,"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)
```

```
diff_ = coefficients(mod.y.3)$worker_id$genre[1]
x1 = as.matrix(set$points$genre$creative)[,"SVD2"]
x2 = as.matrix(set$points$genre$argumentative)[,"SVD2"]

cohensd(diff_,x1,x2)
```

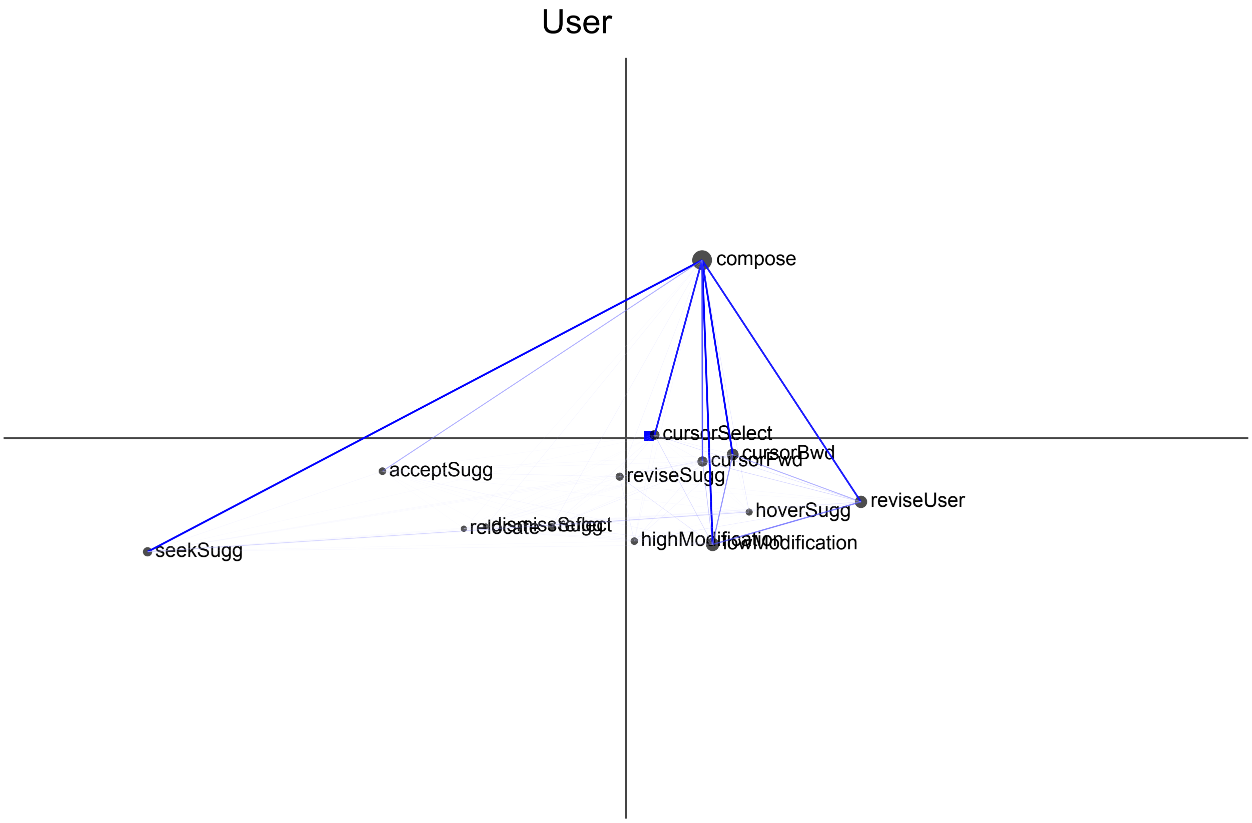
```
#> [1] -0.2227939
```

Effect size (genre)

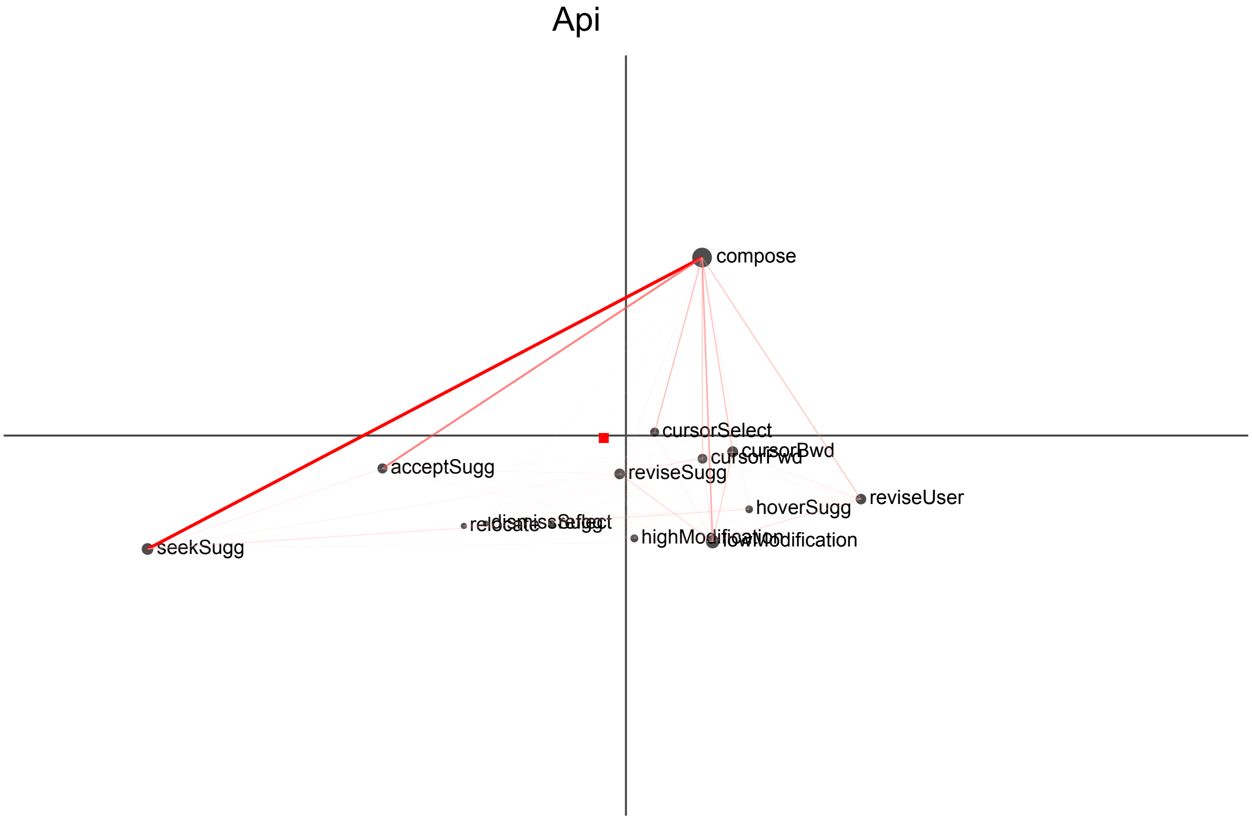
ENA plots

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
```

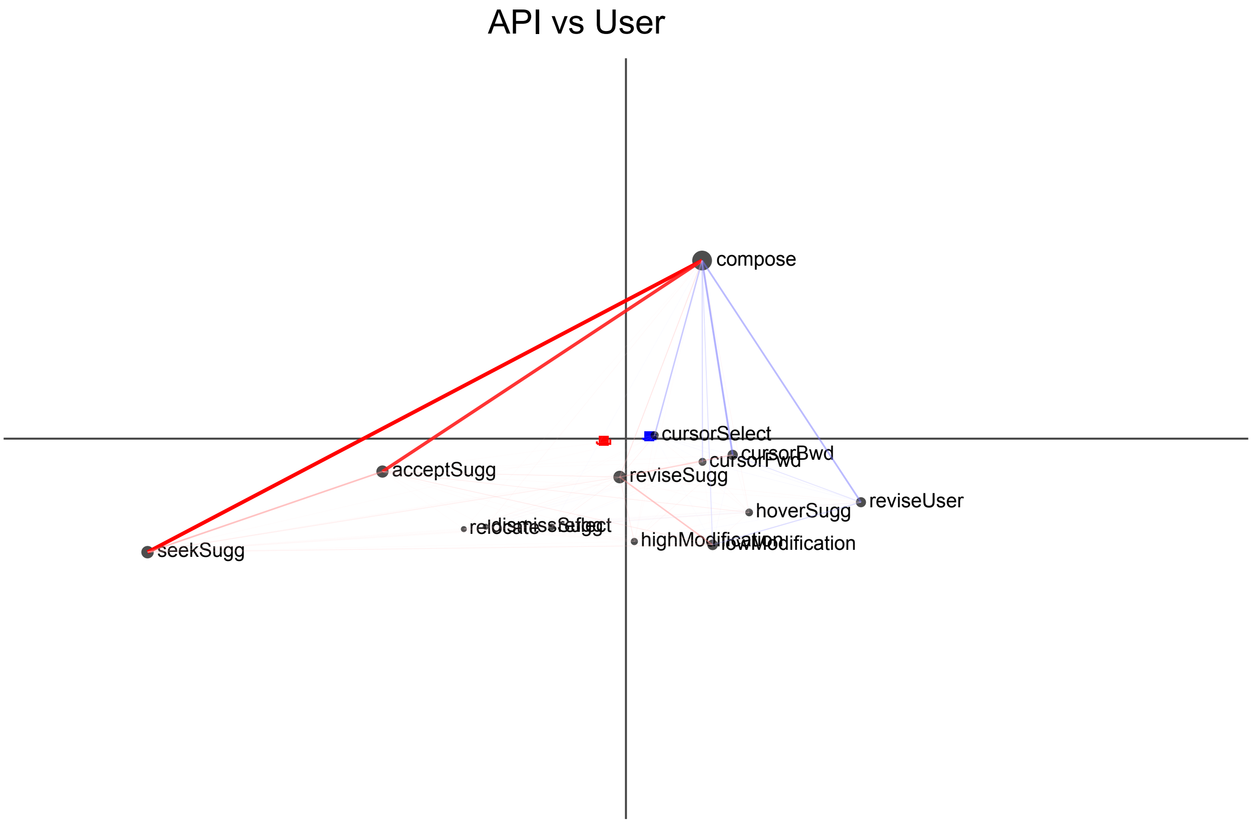


```r
plot_api$plot
```



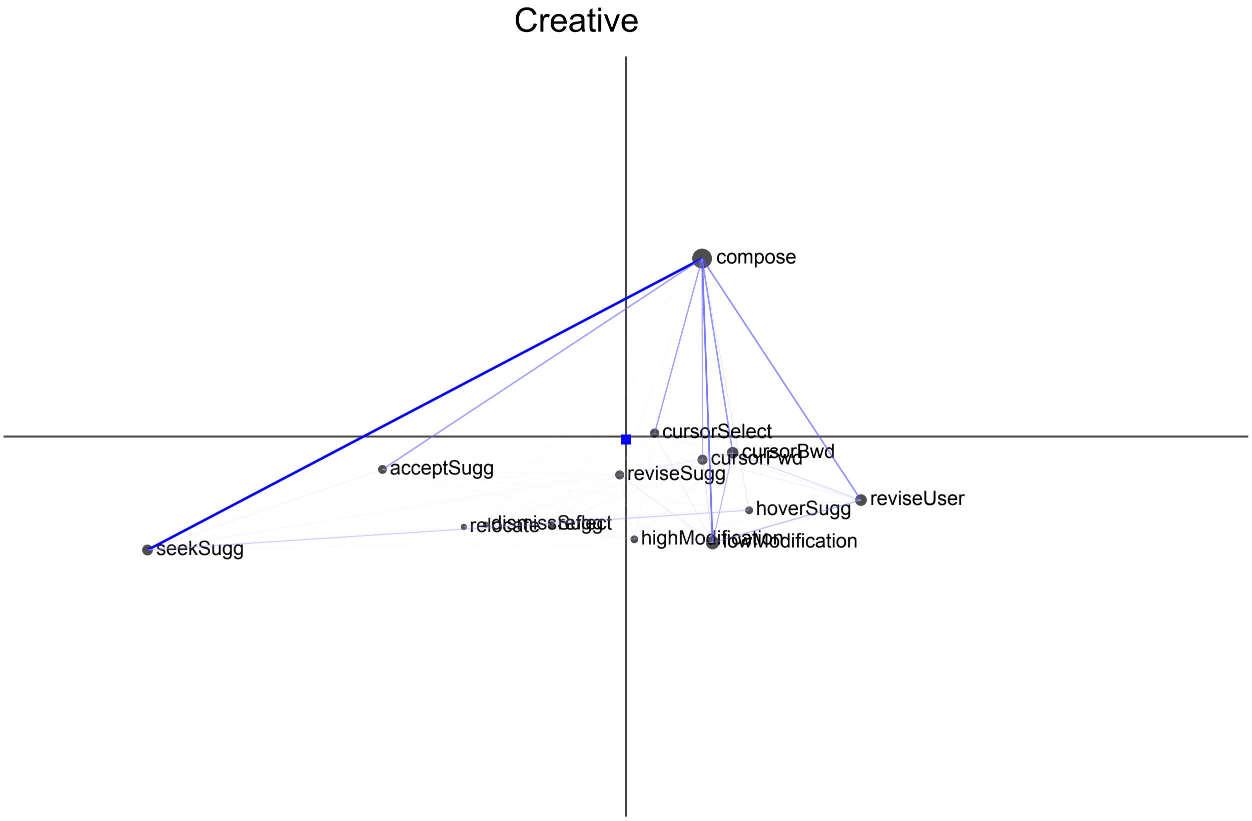
## Network subtraction (ownership)

```
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
```

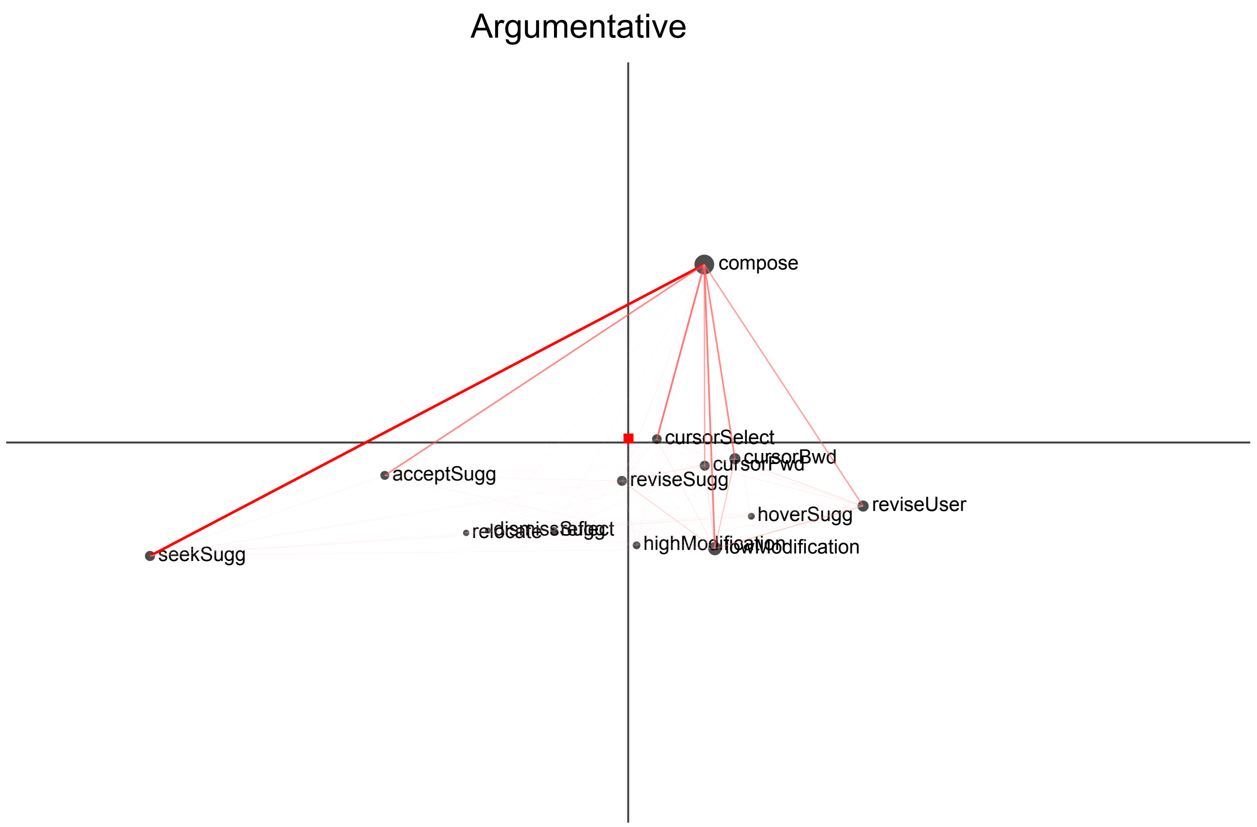


## Mean networks (genre)

```
creat_pts = as.matrix(set$points$genre$creative)
arg_pts = as.matrix(set$points$genre$argumentative)

creat_net = colMeans(as.matrix(set$line.weights$genre$creative))
arg_net = colMeans(as.matrix(set$line.weights$genre$argumentative))

plot_cr = ena.plot(set, scale.to = "network", title = "Creative")  %>%
       #ena.plot.points(points = creat_pts, colors = c("blue"))  %>%
       ena.plot.group(point = creat_pts,
                      colors =c("blue"), confidence.interval = "none")  %>%
       ena.plot.network(network = creat_net,colors = c("blue") )


plot_arg = ena.plot(set, scale.to = "network", title = "Argumentative")  %>%
      # ena.plot.points(points = arg_pts, colors = c("red"))  %>%
       ena.plot.group(point = arg_pts,
                      colors =c("red"), confidence.interval = "none")  %>%
       ena.plot.network(network = arg_net,colors = c("red") )

plot_cr$plot
```

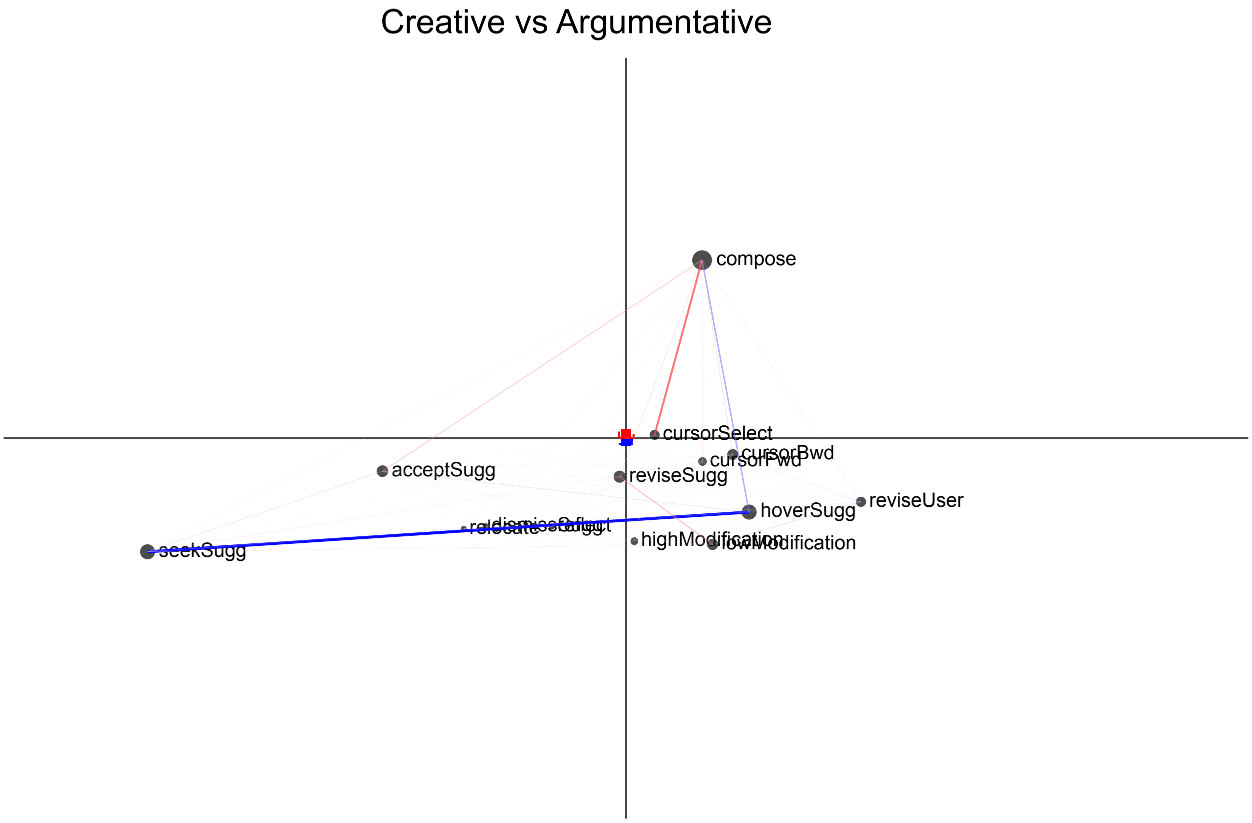


```
plot_arg$plot
```



Network subtraction (genre)

```
net_mult = 5

plot_sub = ena.plot(set, scale.to = "network", title = "Creative vs Argumentative")  %>%
       ena.plot.group(point = creat_pts,
                      colors =c("blue"), confidence.interval = "box")  %>%
       ena.plot.group(point = arg_pts,
                      colors =c("red"), confidence.interval = "box") %>%
       ena.plot.network(network = (creat_net - arg_net) * net_mult,colors = c("blue","red") )
plot_sub$plot
```


Creative vs Argumentative
compose
cursorSelect
cursorBwd
cursorFwd
acceptSugg
reviseSugg
reviseUser
hoverSugg
relocate
dismissSugg
seekSugg
highModification
lowModification