

# Report on data analysis of Data Analysis of Experiments

*Vinh Dang*

*4 November 2015*

## Summary

Summary of all findings in the data analysis:

- Analyze by games \*\* People send more in ID, Score and Combine Games than Simple Game. The behavior of 3 info games are similar. \*\* People send back more in ID, Score and Combine Games than Simple Game. \*\* Profit of ID, Score and Combine Games are much higher than Simple Game.
- Analyze by information showed to users \*\* Showing trust and showing ID let people send more than not showing
- Group effect on game position \*\* There is group interaction on game position, but there is no effect of game position on behavior.
- Group effect on behavior \*\* There is no group effect on showing trust and ID. \*\* The effect of showing Trust in Sender is less than in Receiver \*\* The effect of showing Trust and ID is significant in Profit
- Effect by group \*\* There is no special thing
- Predicting the behavior \*\* Using two trust scores (by own player and partner), we can predict well the behavior of partner
- Group interaction on predicting power \*\* There is Group interaction on predicting power
- Comparing data \*\* The Simple Game is consistency with Giangiacomo \*\* Dubois's data is not useful, there is no effect of treatment, but there is for group interaction.
- Behavior over time \*\* There is no evidence to say that behavior decrease over time.
- Questionnaire analysis \*\* There is no consistency between questionnaire and real data.

## Data Preparation

All data have been collected by running experiments from zTree and zLeaf 3.4.7.

There are 5 experiments so far, and both data files (XLS) and questionnaire files (SBJ) have been put in the directory called “all\_data”. Through the document, the term “experiment” (5 experiments) and group (5 groups) are used interchangeably.

Read data files into R:

```
source("ProcessData.R")
zTT <- readMultiXLS("./all_data")
SBJs <- processMultiSBJ("./all_data")

# Create my sending proportional
zTT[2]$subjects$my_send_proportional <- ifelse(zTT[2]$subjects$Type ==
  0, zTT[2]$subjects$Contribution/10, ifelse(zTT[2]$subjects$PartnerDecision >
  0, zTT[2]$subjects$Contribution/3/zTT[2]$subjects$PartnerDecision,
  -1))

zTT[2]$subjects$CurrGameProfit <- ifelse(zTT[2]$subjects$Type ==
  0, zTT[2]$subjects$PartnerDecision - zTT[2]$subjects$Contribution,
  zTT[2]$subjects$PartnerDecision * 3 - zTT[2]$subjects$Contribution)
```

## Data analysis

In this section, we presented how to analyze data from XLS files

### Data preparation

First, we define experience parameters

```
# Numbers of user of a group
num_users = 6
# Number of rounds each user play to each other
average_rounds = 5
# Number of games for each group
num_games = 4
# Number of rounds for each game (should be 25)
num_rounds_per_game = (num_users - 1) * average_rounds
# Number of rounds for each experiment (should be
# 100, because we have 4 games)
num_rounds_per_exp = num_rounds_per_game * num_games
# Number of experiments (it is 5 at the time of
# writing, but can be increased if we organize more
# experiments)
num_exp = nrow(zTT[1]$globals)/num_rounds_per_exp

Type_names = c("SENDER", "RECEIVER")
SIMPLE_GAME_ORDERS = c(3, 2, 4, 1, 2)
ID_GAME_ORDERS = c(1, 4, 1, 2, 3)
SCORE_GAME_ORDERS = c(2, 1, 3, 4, 4)
COMBINE_GAME_ORDERS = c(4, 3, 2, 3, 1)
```

```
GAME_NAMES <- c("Simple GAME", "ID GAME", "Score GAME",  
               "Combine GAME")
```

We also define some global variables which will be used later.

Then, we read through all the data to arrange the game, because during the experiment, the order of games has been shuffled.

```
# first, create empty data frames to hold all the  
# particular games  
simple_games <- zTT[2]$subjects[0, ]  
id_games <- zTT[2]$subjects[0, ]  
score_games <- zTT[2]$subjects[0, ]  
combine_games <- zTT[2]$subjects[0, ]
```

Go through each experiment, find the order of the game, and put to the corresponding data frame we created right above.

From now, we have four data frames which contains all data for four games.

## Basic analysis

We can compute some basic metrics as follow:

For this, we can see that, the total behavior for 3 games with information are similar, and much more better than simple game.

```
# calculate game metrics calculate average sending
# amount per game
print("Mean of sending amount per games")

## [1] "Mean of sending amount per games"
print(mean(simple_games$Contribution))

## [1] 2.64
print(mean(id_games$Contribution))

## [1] 6.242667
print(mean(score_games$Contribution))

## [1] 6.398667
print(mean(combine_games$Contribution))

## [1] 6.726667
print("Mean of sending amount by sender per games")

## [1] "Mean of sending amount by sender per games"
print(mean(simple_games[simple_games$Type == 0, ]$Contribution))

## [1] 3.002667
print(mean(id_games[id_games$Type == 0, ]$Contribution))

## [1] 5.304
print(mean(score_games[score_games$Type == 0, ]$Contribution))

## [1] 5.288
print(mean(combine_games[combine_games$Type == 0, ]$Contribution))

## [1] 5.453333
print(mean(simple_games[simple_games$Type == 1, ]$Contribution))

## [1] 2.277333
print(mean(id_games[id_games$Type == 1, ]$Contribution))

## [1] 7.181333
print(mean(score_games[score_games$Type == 1, ]$Contribution))

## [1] 7.509333
print(mean(combine_games[combine_games$Type == 1, ]$Contribution))

## [1] 8
```

```

# calculate average sending proportion per game
# print ('Mean of sending proportion per game')
# print (mean
# (simple_games[simple_games$my_send_proportional
# >= 0,]$my_send_proportional)) print (mean
# (id_games[id_games$my_send_proportional >=
# 0,]$my_send_proportional)) print (mean
# (score_games[score_games$my_send_proportional >=
# 0,]$my_send_proportional)) print (mean
# (combine_games[combine_games$my_send_proportional
# >= 0,]$my_send_proportional))

print("Average of sending amount by senders per game")

## [1] "Average of sending amount by senders per game"
print(mean(simple_games[simple_games$my_send_proportional >=
  0 & simple_games$Type == 0, ]$my_send_proportional))

## [1] 0.3002667
print(mean(id_games[id_games$my_send_proportional >=
  0 & id_games$Type == 0, ]$my_send_proportional))

## [1] 0.5304
print(mean(score_games[score_games$my_send_proportional >=
  0 & score_games$Type == 0, ]$my_send_proportional))

## [1] 0.5288
print(mean(combine_games[combine_games$my_send_proportional >=
  0 & combine_games$Type == 0, ]$my_send_proportional))

## [1] 0.5453333
print("Average of sending back by receiver per game")

## [1] "Average of sending back by receiver per game"
print(mean(simple_games[simple_games$my_send_proportional >=
  0 & simple_games$Type == 1, ]$my_send_proportional))

## [1] 0.2615905
print(mean(id_games[id_games$my_send_proportional >=
  0 & id_games$Type == 1, ]$my_send_proportional))

## [1] 0.4409586
print(mean(score_games[score_games$my_send_proportional >=
  0 & score_games$Type == 1, ]$my_send_proportional))

## [1] 0.476037
print(mean(combine_games[combine_games$my_send_proportional >=
  0 & combine_games$Type == 1, ]$my_send_proportional))

## [1] 0.4765017

```

```

print("Average profit user get in 1 round per game")

## [1] "Average profit user get in 1 round per game"
print(mean(simple_games[simple_games$Type == 0, ]$CurrGameProfit))

## [1] -0.7253333
print(mean(id_games[id_games$Type == 0, ]$CurrGameProfit))

## [1] 1.877333
print(mean(score_games[score_games$Type == 0, ]$CurrGameProfit))

## [1] 2.221333
print(mean(combine_games[combine_games$Type == 0, ]$CurrGameProfit))

## [1] 2.546667
print(mean(simple_games[simple_games$Type == 1, ]$CurrGameProfit))

## [1] 6.730667
print(mean(id_games[id_games$Type == 1, ]$CurrGameProfit))

## [1] 8.730667
print(mean(score_games[score_games$Type == 1, ]$CurrGameProfit))

## [1] 8.354667
print(mean(combine_games[combine_games$Type == 1, ]$CurrGameProfit))

## [1] 8.36
print("Response time for games")

## [1] "Response time for games"
print(mean(simple_games[simple_games$Type == 0, ]$response_time))

## [1] 9.874856
print(mean(simple_games[simple_games$Type == 1, ]$response_time))

## [1] 12.73053
print(mean(id_games[id_games$Type == 0, ]$response_time))

## [1] 11.03453
print(mean(id_games[id_games$Type == 1, ]$response_time))

## [1] 15.34007
print(mean(score_games[score_games$Type == 0, ]$response_time))

## [1] 10.90826
print(mean(score_games[score_games$Type == 1, ]$response_time))

## [1] 15.04132

```

```
print(mean(combine_games[combine_games$Type == 0, ]$response_time))
```

```
## [1] 12.35263
```

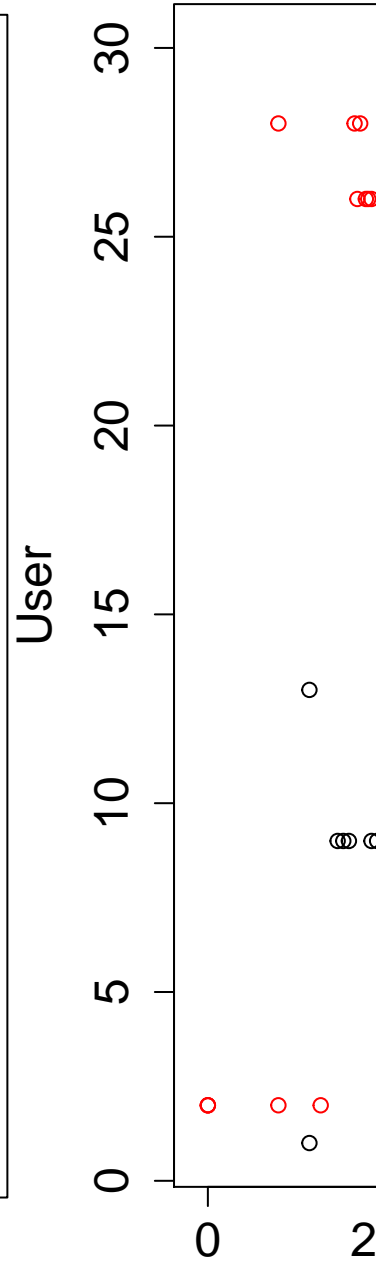
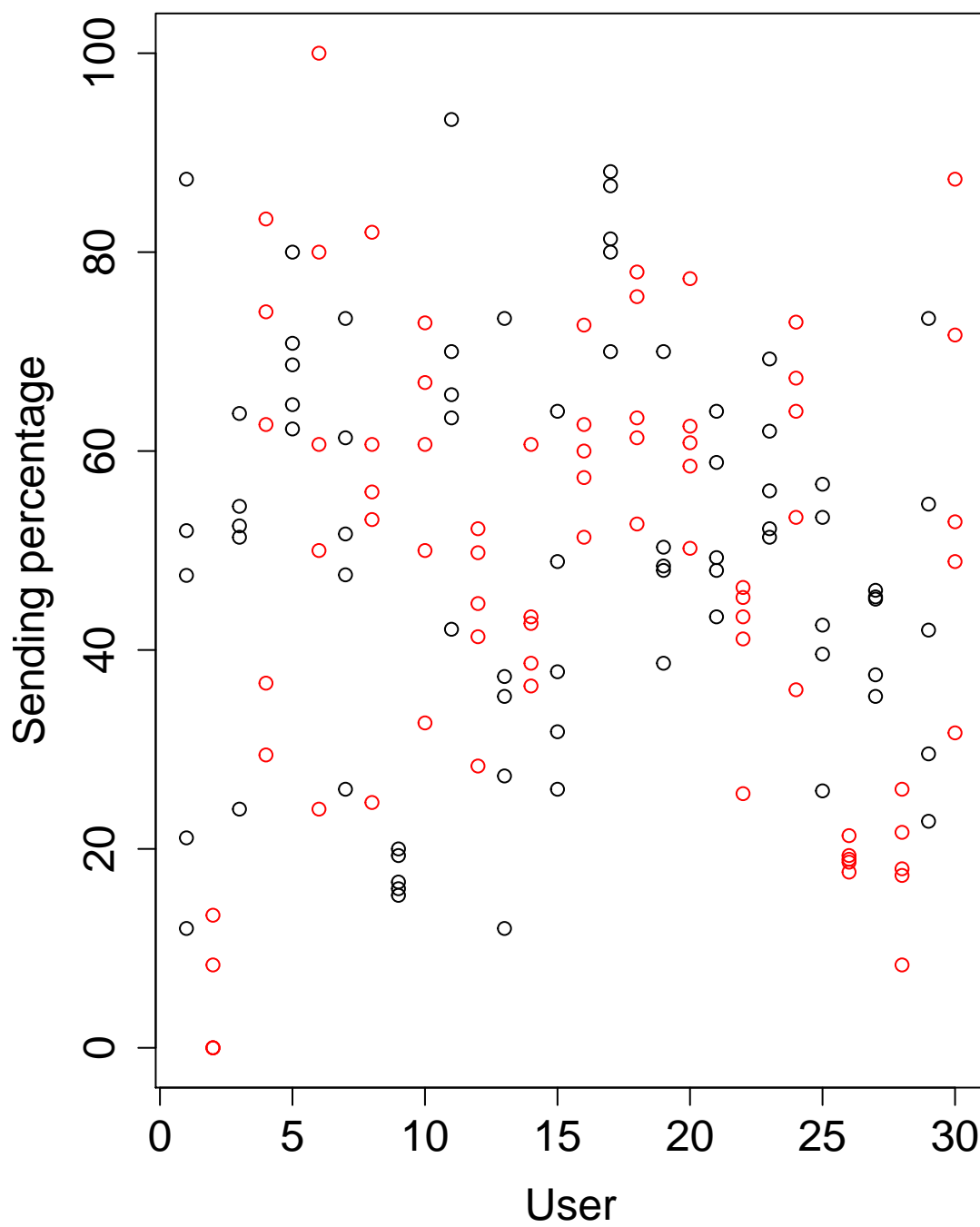
```
print(mean(combine_games[combine_games$Type == 1, ]$response_time))
```

```
## [1] 17.29802
```

Of course, these above metrics are very basic one. We need to calculate the group interaction on SHOW\_TRUST and SHOW\_ID variables (both of them are boolean values, mean in a particular game, we show trust score or identity to users or not). In order to do this, we will use 2 - way and 3 - way ANOVA in wide format. More details at <http://www.uni-kiel.de/psychologie/rexrepos/posts/anovaSPFpqr.html>

Testing the difference of users' behavior on different partners

## [1] 0.384607





### **Game metrics for each person of each group**

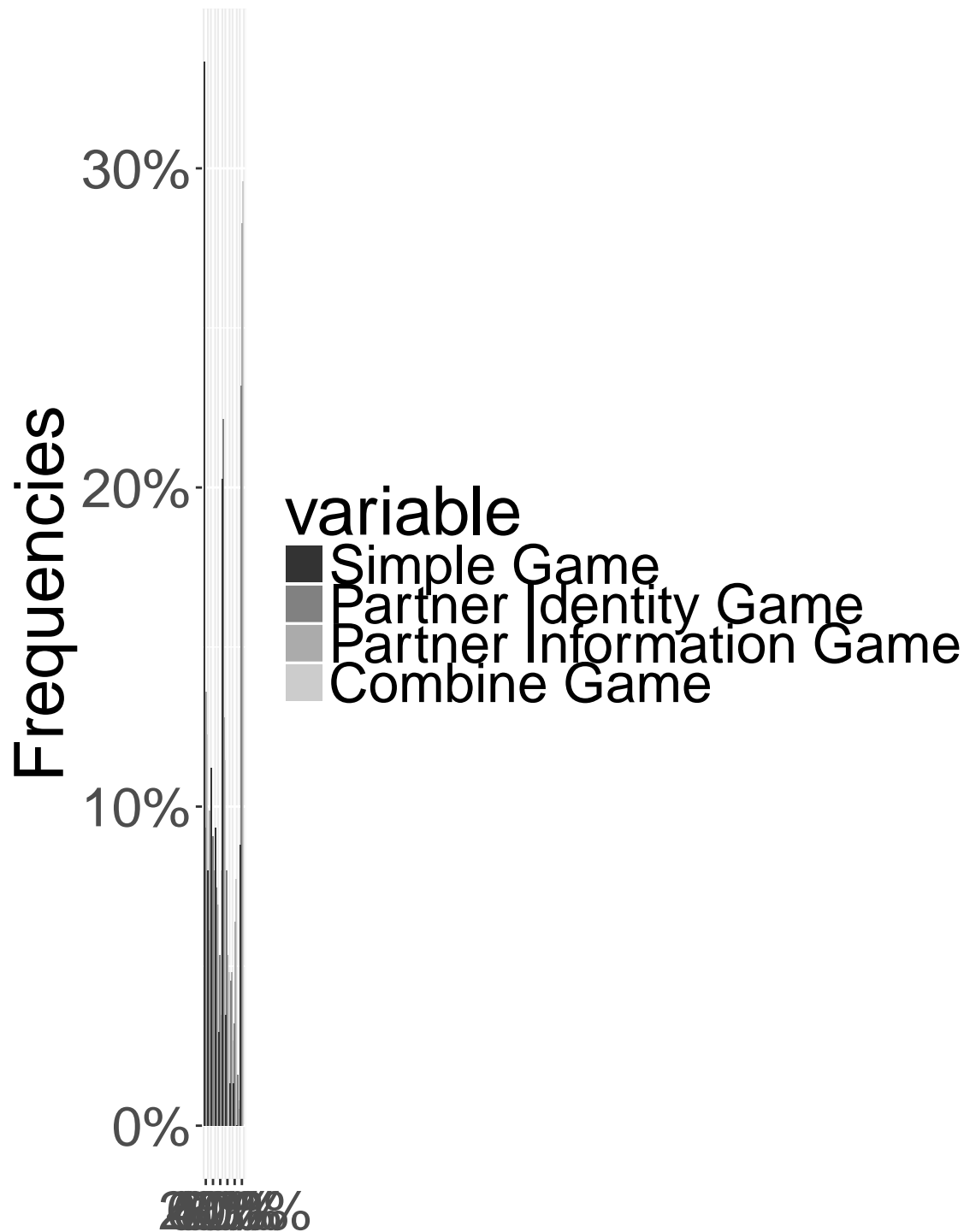
We calculate behavior for each group and see the effect of showing trust or showing ID. We can see that it is really better to show the information to users.

Temporary remove, move to effect of each group.

```

require(ggplot2)
require(reshape2)
require(scales)
c <- cbind(simple_games[simple_games$Type == 0 & simple_games$my_send_proportional >=
0, ]$Contribution, id_games[id_games$Type == 0 &
id_games$my_send_proportional >= 0, ]$Contribution,
score_games[score_games$Type == 0 & score_games$my_send_proportional >=
0, ]$Contribution, combine_games[combine_games$Type ==
0 & combine_games$my_send_proportional >= 0,
]$Contribution)
c = c/10
d <- as.data.frame(c)
colnames(d) <- c("Simple Game", "Partner Identity Game",
"Partner Information Game", "Combine Game")
# make histogram of sending amount by senders
ggplot(melt(d), aes(value, fill = variable)) + geom_histogram(position = "dodge",
binwidth = 0.1, aes(y = ..count../sum(..count..) *
4)) + scale_x_continuous(breaks = seq(0, 1,
0.2), labels = percent) + xlab("Sending percentage") +
scale_y_continuous(labels = percent) + ylab("Frequencies") +
scale_fill_grey() + theme(text = element_text(size = 30))

```



```
# For receivers
c <- cbind(simple_games[simple_games$Type == 1 & simple_games$my_send_proportional >=
0, ]$my_send_proportional, id_games[id_games$Type ==
1 & id_games$my_send_proportional >= 0, ]$my_send_proportional,
score_games[score_games$Type == 1 & score_games$my_send_proportional >=
0, ]$my_send_proportional, combine_games[combine_games$Type ==
```

```

    1 & combine_games$my_send_proportional >= 0,
    ]$my_send_proportional)
d <- as.data.frame(c)
colnames(d) <- c("Simple Game", "Partner Identity Game",
  "Partner Information Game", "Combine Game")
# make histogram of sending amount by senders
ggplot(melt(d), aes(value, fill = variable)) + geom_histogram(position = "dodge",
  binwidth = 0.1, aes(y = ..count../sum(..count..) *
    4)) + scale_x_continuous(breaks = seq(0, 1,
  0.2), labels = percent) + xlab("Sending percentage") +
  scale_y_continuous(labels = percent) + scale_fill_grey() +
  theme(text = element_text(size = 30)) + ylab("")

```



```

    list(sg_senders$Date, sg_senders$Subject),
    mean)

pdg_senders <- id_games[id_games$Type == type &
  id_games$my_send_proportional >= 0, ]
pdg_senders_avg <- aggregate(pdg_senders$my_send_proportional,
  list(pdg_senders$Date, pdg_senders$Subject),
  mean)

pfg_senders <- score_games[score_games$Type ==
  type & score_games$my_send_proportional >=
  0, ]
pfg_senders_avg <- aggregate(pfg_senders$my_send_proportional,
  list(pfg_senders$Date, pfg_senders$Subject),
  mean)

cg_senders <- combine_games[combine_games$Type ==
  type & combine_games$my_send_proportional >=
  0, ]
cg_senders_avg <- aggregate(cg_senders$my_send_proportional,
  list(cg_senders$Date, cg_senders$Subject),
  mean)

perml_ks(sg_senders_avg[["x"]], pdg_senders_avg[["x"]])
perml_ks(sg_senders_avg[["x"]], pfg_senders_avg[["x"]])
perml_ks(sg_senders_avg[["x"]], cg_senders_avg[["x"]])

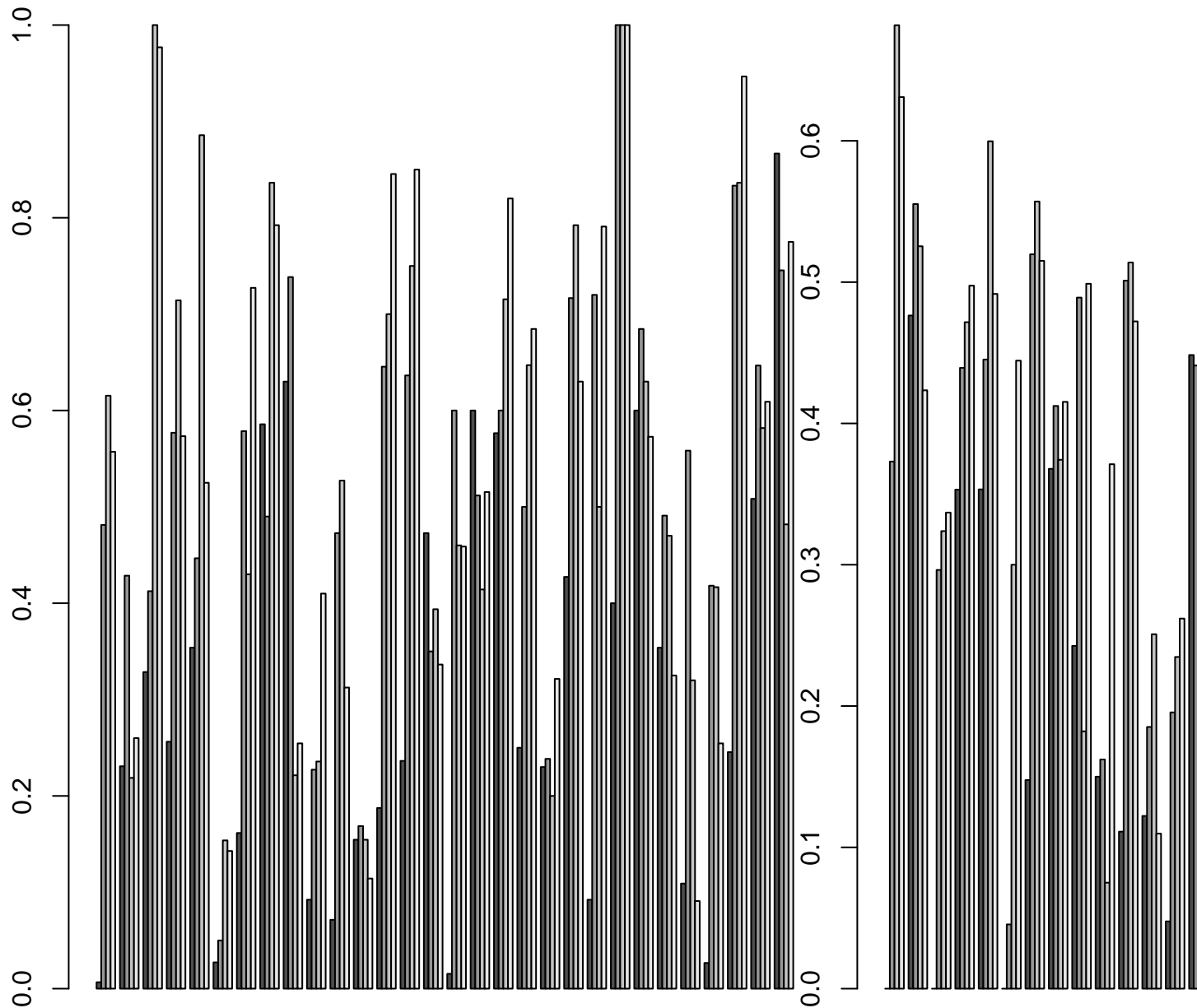
perml_ks(pdg_senders_avg[["x"]], pfg_senders_avg[["x"]])
perml_ks(pdg_senders_avg[["x"]], cg_senders_avg[["x"]])

perml_ks(pfg_senders_avg[["x"]], cg_senders_avg[["x"]])

c2 <- cbind(sg_senders_avg[["x"]], pdg_senders_avg[["x"]],
  pfg_senders_avg[["x"]], cg_senders_avg[["x"]])
d2 <- as.data.frame(c2)
colnames(d2) <- c("Simple Game", "Partner Identity Game",
  "Partner Information Game", "Combine Game")

r2 <- rbind(sg_senders_avg[["x"]], pdg_senders_avg[["x"]],
  pfg_senders_avg[["x"]], cg_senders_avg[["x"]])
barplot(r2, beside = TRUE)
}

```



```
# plot (id, sg_senders_avg[['x']], ylim = c(0,1),
# col='red', main = 'Sending percentage of
# senders', xlab = 'Participant', ylab = 'Sending
# percentage', yaxt='n') axis(2,
# at=pretty(sg_senders_avg[['x']]), lab= paste
# (pretty(sg_senders_avg[['x']]) * 100, '%'),
# las=TRUE) lines (id, sg_senders_avg[['x']], lty =
# 2, col='red') lines (id, pdg_senders_avg[['x']],
# lty = 2, col='blue', type = 'o') lines (id,
# pfg_senders_avg[['x']], lty = 2, col='green',
# type = 'o') lines (id, cg_senders_avg[['x']], lty
# = 2, col='purple', type = 'o')
```

Compare behavior between games



## Compare between games using ANOVA

Before, we compared average of sending amount and average of sending proportion between games using pairwise t-test on all possible pair games.

However, we can use ANOVA to test all four games in once. More information at <http://brownmath.com/stat/anova1>.

The code is taken from <http://www.sthda.com/english/wiki/one-way-anova-test-in-r>

```
# For sender
send_simple = simple_games[simple_games$Type == 0,
  ]$my_send_proportional
send_id = id_games[id_games$Type == 0, ]$my_send_proportional
send_score = score_games[score_games$Type == 0, ]$my_send_proportional
send_combine = combine_games[combine_games$Type ==
  0, ]$my_send_proportional
dati = c(send_simple, send_id, send_score, send_combine)
groups = factor(rep(c("simple", "id", "score", "combine"),
  each = length(send_simple)))

# test variance homogeneity
bartlett.test(dati, groups)

##
## Bartlett test of homogeneity of variances
##
## data:  dati and groups
## Bartlett's K-squared = 18.713, df = 3, p-value = 0.0003135

fligner.test(dati, groups)

##
## Fligner-Killeen test of homogeneity of variances
##
## data:  dati and groups
## Fligner-Killeen:med chi-squared = 36.256, df = 3, p-value =
## 6.61e-08

# ANOVA ANOVA answers if four means are equal or
# not
fit = lm(formula = dati ~ groups)

anova(fit)

## Analysis of Variance Table
##
## Response: dati
##          Df Sum Sq Mean Sq F value    Pr(>F)
## groups    3  15.539   5.1795  43.076 < 2.2e-16 ***
## Residuals 1496 179.882   0.1202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# TukeyHSD test, to answer which mean is greater
# than which mean
TukeyHSD(aov(fit))
```

```

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = fit)
##
## $groups
##           diff      lwr      upr    p adj
## id-combine  -0.01493333 -0.08006388  0.05019721 0.9352161
## score-combine -0.01653333 -0.08166388  0.04859721 0.9145584
## simple-combine -0.24506667 -0.31019721 -0.17993612 0.0000000
## score-id      -0.00160000 -0.06673055  0.06353055 0.9999095
## simple-id     -0.23013333 -0.29526388 -0.16500279 0.0000000
## simple-score  -0.22853333 -0.29366388 -0.16340279 0.0000000

# Using multcomp package, little bit stronger than
# TukeyHSD test
library(multcomp)
pairwise <- glht(fit, linfct = mcp(groups = "Tukey"))
summary(pairwise)

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = dati ~ groups)
##
## Linear Hypotheses:
##           Estimate Std. Error t value Pr(>|t|)
## id - combine == 0   -0.01493    0.02532  -0.590    0.935
## score - combine == 0 -0.01653    0.02532  -0.653    0.915
## simple - combine == 0 -0.24507    0.02532 -9.677 <1e-05 ***
## score - id == 0     -0.00160    0.02532  -0.063    1.000
## simple - id == 0     -0.23013    0.02532 -9.088 <1e-05 ***
## simple - score == 0  -0.22853    0.02532 -9.024 <1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

# Kruskal - Wallis test for non-parametric
kruskal.test(dati ~ groups)

##
## Kruskal-Wallis rank sum test
##
## data:  dati by groups
## Kruskal-Wallis chi-squared = 121.97, df = 3, p-value < 2.2e-16

# if K-W test is significant, a post-hoc test can
# be performed here we use Dunn test
# http://rcompanion.org/rcompanion/d_06.html
library(FSA)
PT = dunnTest(dati ~ groups)
PT

##           Comparison      Z      P.unadj      P.adj

```

```
## 1      combine - id 0.1421786 8.869389e-01 8.869389e-01
## 2  combine - score 0.5474346 5.840802e-01 1.000000e+00
## 3      id - score 0.4052560 6.852894e-01 1.000000e+00
## 4  combine - simple 9.2352564 2.576682e-20 1.546009e-19
## 5      id - simple 9.0930777 9.627437e-20 4.813718e-19
## 6  score - simple 8.6878218 3.694549e-18 1.477819e-17

# using Nemenyi test not suitable for groups with
# different length (Zar, J.H. 2010. Biostatistical
# Analysis, 5th ed. Pearson Prentice Hall: Upper
# Saddle River, NJ.)
library(DescTools)
nt = NemenyiTest(x = dati, g = groups, dist = "tukey")
nt

##
## Nemenyi's test of multiple comparisons for independent samples (tukey)
##
##           mean.rank.diff      pval
## id-combine          -4.445333 0.9990
## score-combine       -17.116000 0.9490
## simple-combine     -288.748000 4.9e-14 ***
## score-id           -12.670667 0.9783
## simple-id          -284.302667 2.7e-14 ***
## simple-score       -271.632000 3.6e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# For sender: net sending amount
send_simple = simple_games[simple_games$Type == 0,
  ]$Contribution
send_id = id_games[id_games$Type == 0, ]$Contribution
send_score = score_games[score_games$Type == 0, ]$Contribution
send_combine = combine_games[combine_games$Type ==
  0, ]$Contribution
dati = c(send_simple, send_id, send_score, send_combine)
groups = factor(rep(c("simple", "id", "score", "combine"),
  each = length(send_simple)))

# test variance homogeneity
bartlett.test(dati, groups)

##
## Bartlett test of homogeneity of variances
##
## data:  dati and groups
## Bartlett's K-squared = 18.713, df = 3, p-value = 0.0003135

fligner.test(dati, groups)

##
## Fligner-Killeen test of homogeneity of variances
##
## data:  dati and groups
## Fligner-Killeen:med chi-squared = 34.209, df = 3, p-value =
## 1.79e-07
```

```

# ANOVA ANOVA answers if four means are equal or
# not
fit = lm(formula = dati ~ groups)

anova(fit)

## Analysis of Variance Table
##
## Response: dati
##           Df Sum Sq Mean Sq F value    Pr(>F)
## groups      3  1553.9   517.95  43.076 < 2.2e-16 ***
## Residuals 1496  17988.2    12.02
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# TukeyHSD test, to answer which mean is greater
# than which mean
TukeyHSD(aov(fit))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = fit)
##
## $groups
##           diff      lwr      upr    p adj
## id-combine   -0.1493333 -0.8006388  0.5019721 0.9352161
## score-combine -0.1653333 -0.8166388  0.4859721 0.9145584
## simple-combine -2.4506667 -3.1019721 -1.7993612 0.0000000
## score-id      -0.0160000 -0.6673055  0.6353055 0.9999095
## simple-id     -2.3013333 -2.9526388 -1.6500279 0.0000000
## simple-score  -2.2853333 -2.9366388 -1.6340279 0.0000000

# For receiver
receive_simple = simple_games[simple_games$Type ==
  1 & simple_games$my_send_proportional >= 0, ]$my_send_proportional
receive_id = id_games[id_games$Type == 1 & id_games$my_send_proportional >=
  0, ]$my_send_proportional
receive_score = score_games[score_games$Type == 1 &
  score_games$my_send_proportional >= 0, ]$my_send_proportional
receive_combine = combine_games[combine_games$Type ==
  1 & combine_games$my_send_proportional >= 0, ]$my_send_proportional
dati = c(receive_simple, receive_id, receive_score,
  receive_combine)
groups = factor(c(rep("simple", length(receive_simple)),
  rep("id", length(receive_id)), rep("score", length(receive_score)),
  rep("combine", length(receive_combine))))

# test variance homogeneity
bartlett.test(dati, groups)

##
## Bartlett test of homogeneity of variances
##
## data:  dati and groups

```

```
## Bartlett's K-squared = 20.316, df = 3, p-value = 0.000146
fligner.test(dati, groups)

##
## Fligner-Killeen test of homogeneity of variances
##
## data:  dati and groups
## Fligner-Killeen:med chi-squared = 57.124, df = 3, p-value =
## 2.418e-12
# ANOVA ANOVA answers if four means are equal or
# not
fit = lm(formula = dati ~ groups)

anova(fit)

## Analysis of Variance Table
##
## Response: dati
##           Df Sum Sq Mean Sq F value    Pr(>F)
## groups      3  8.476  2.82527   53.889 < 2.2e-16 ***
## Residuals 1239 64.957  0.05243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# TukeyHSD test, to answer which mean is greater
# than which mean
TukeyHSD(aov(fit))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = fit)
##
## $groups
##           diff          lwr          upr      p adj
## id-combine -0.0355430588 -0.08109565  0.01000953 0.1857852
## score-combine -0.0004646928 -0.04656710  0.04563771 0.9999937
## simple-combine -0.2149111883 -0.26433187 -0.16549050 0.0000000
## score-id       0.0350783661 -0.01065251  0.08080924 0.1986695
## simple-id     -0.1793681295 -0.22844241 -0.13029385 0.0000000
## simple-score  -0.2144464955 -0.26403156 -0.16486143 0.0000000
# Using multcomp package, little bit stronger than
# TukeyHSD test
library(multcomp)
pairwise <- glht(fit, linfct = mcp(groups = "Tukey"))
summary(pairwise)

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = dati ~ groups)
```

```
##
## Linear Hypotheses:
##               Estimate Std. Error t value Pr(>|t|)
## id - combine == 0    -0.0355431  0.0177074  -2.007    0.185
## score - combine == 0  -0.0004647  0.0179211  -0.026    1.000
## simple - combine == 0 -0.2149112  0.0192110 -11.187 <0.001 ***
## score - id == 0       0.0350784  0.0177767   1.973    0.198
## simple - id == 0      -0.1793681  0.0190763  -9.403 <0.001 ***
## simple - score == 0   -0.2144465  0.0192749 -11.126 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
# Kruskal - Wallis test for non-parametric
kruskal.test(dati ~ groups)
```

```
##
## Kruskal-Wallis rank sum test
##
## data:  dati by groups
## Kruskal-Wallis chi-squared = 128.98, df = 3, p-value < 2.2e-16
```

```
# if K-W test is significant, a post-hoc test can
# be performed here we use Dunn test
# http://rcompanion.org/rcompanion/d\_06.html
```

```
library(FSA)
PT = dunnTest(dati ~ groups)
PT
```

```
##           Comparison           Z      P.unadj      P.adj
## 1 combine - id  1.7719928 7.639575e-02 1.527915e-01
## 2 combine - score -0.3485123 7.274555e-01 7.274555e-01
## 3 id - score -2.1164283 3.430840e-02 1.029252e-01
## 4 combine - simple  9.8786197 5.153790e-23 2.576895e-22
## 5 id - simple  8.3035197 1.010716e-16 4.042864e-16
## 6 score - simple 10.1699055 2.701706e-24 1.621024e-23
```

```
# For receiver for net amount
receive_simple = simple_games[simple_games$Type ==
  1 & simple_games$my_send_proportional >= 0, ]$Contribution
receive_id = id_games[id_games$Type == 1 & id_games$my_send_proportional >=
  0, ]$Contribution
receive_score = score_games[score_games$Type == 1 &
  score_games$my_send_proportional >= 0, ]$Contribution
receive_combine = combine_games[combine_games$Type ==
  1 & combine_games$my_send_proportional >= 0, ]$Contribution
dati = c(receive_simple, receive_id, receive_score,
  receive_combine)
groups = factor(c(rep("simple", length(receive_simple)),
  rep("id", length(receive_id)), rep("score", length(receive_score)),
  rep("combine", length(receive_combine))))
```

```
# test variance homogeneity
bartlett.test(dati, groups)
```

```
##
## Bartlett test of homogeneity of variances
```

```
##
## data:  dati and groups
## Bartlett's K-squared = 65.052, df = 3, p-value = 4.89e-14
fligner.test(dati, groups)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  dati and groups
## Fligner-Killeen:med chi-squared = 64.215, df = 3, p-value =
## 7.382e-14
# ANOVA ANOVA answers if four means are equal or
# not
fit = lm(formula = dati ~ groups)

anova(fit)

## Analysis of Variance Table
##
## Response: dati
##           Df Sum Sq Mean Sq F value    Pr(>F)
## groups      3   5550  1850.11   52.442 < 2.2e-16 ***
## Residuals 1239   43711    35.28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# TukeyHSD test, to answer which mean is greater
# than which mean
TukeyHSD(aov(fit))

##  Tukey multiple comparisons of means
##    95% family-wise confidence level
##
## Fit: aov(formula = fit)
##
## $groups
##           diff          lwr          upr      p adj
## id-combine  -1.1979528 -2.3796191 -0.0162865 0.0454951
## score-combine -0.4271830 -1.6231118  0.7687458 0.7947613
## simple-combine -5.7025410 -6.9845484 -4.4205336 0.0000000
## score-id       0.7707698 -0.4155213  1.9570609 0.3392931
## simple-id     -4.5045882 -5.7776097 -3.2315667 0.0000000
## simple-score  -5.2753580 -6.5616295 -3.9890866 0.0000000

==> all the tests confirmed that Simple_Game < ID-game ~ score-game ~ combine-game
```

## Group effect on game position

```
## [1] "ANOVA analysis with relative sending on GroupID:game_pos"
## [1] "-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)      27.8796      1   2.0236      25 344.4249 3.916e-16 ***
## GroupID          2.0040      4   2.0236      25   6.1893 0.001325 **
## game_setting      0.0789      3   1.7949      75   0.1702 0.914425
## GroupID:game_setting 1.8542     12   1.7949      75   6.4565 9.721e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with absolute profit on GroupID:game_pos"
## [1] "-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)     270.497      1   73.067      25 92.5503 6.962e-10 ***
## GroupID         29.167      4   73.067      25   2.4949 0.06857 .
## game_setting      3.324      3  123.981      75   0.0570 0.98126
## GroupID:game_setting 233.152     12  123.981      75  11.7534 6.777e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with absolute response time on GroupID:game_pos"
## [1] "-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)    14746.3      1  1127.14      25 327.0727 7.157e-16 ***
## GroupID         663.6      4  1127.14      25   3.6799 0.017274 *
## game_setting     799.8      3   625.09      75  10.0497 0.001358 **
## GroupID:game_setting 318.3     12   625.09      75   3.1829 0.001029 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with relative sending on GroupID:game_pos"
## [1] "-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)     20.1800      1   2.1185      25 238.1416 2.757e-14 ***
## GroupID          0.1004      4   2.1185      25   0.2961 0.8777
## game_setting      0.0601      3   1.1328      75   0.3243 0.8078
## GroupID:game_setting 0.7415     12   1.1328      75   4.0907 6.819e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with absolute profit on GroupID:game_pos"
## [1] "-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)     7724.6      1  343.58      25 562.076 < 2.2e-16 ***
## GroupID         676.7      4  343.58      25  12.311 1.146e-05 ***
## game_setting      65.0      3  277.93      75   1.292 0.322
## GroupID:game_setting 201.2     12  277.93      75   4.524 1.930e-05 ***
```



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with absolute response time on GroupID:game_pos"
## [1] "-----*****-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
```

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	27203.7	1	1428.00	25	476.2559	< 2.2e-16 ***
GroupID	1249.8	4	1428.00	25	5.4699	0.002640 **
game_setting	2251.1	3	724.48	75	9.5859	0.001651 **
GroupID:game_setting	939.3	12	724.48	75	8.1037	1.658e-09 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
```

## Test the effect of groups

In the following code chunk, we will use 'Anova' function, which is the part of 'car' package.

We want to test the effect of TRUST and ID to behavior, and see is there group interaction in the data or not.

We can see that for relative sending, there is no group effect.

```
## [1] "ANOVA analysis with relative sending on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)    27.8796      1  2.02363    25 344.4249
## GroupID        2.0040      4  2.02363    25   6.1893
## SHOW_TRUST      0.4955      1  0.91768    25   5.1728
## GroupID:SHOW_TRUST 0.3832      4  0.91768    25   2.6095
## SHOW_ID         0.4132      1  0.48582    25  16.8279
## GroupID:SHOW_ID   0.0982      4  0.48582    25   1.2634
## SHOW_TRUST:SHOW_ID 0.3741      1  0.39138    25   8.8590
## GroupID:SHOW_TRUST:SHOW_ID 0.1689      4  0.39138    25   2.6977
##
##              Pr(>F)
## (Intercept) 3.916e-16 ***
## GroupID     0.001325 **
## SHOW_TRUST  0.085315 .
## GroupID:SHOW_TRUST 0.059743 .
## SHOW_ID     0.014827 *
## GroupID:SHOW_ID 0.310568
## SHOW_TRUST:SHOW_ID 0.040884 *
## GroupID:SHOW_TRUST:SHOW_ID 0.053764 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
## [1] "-----*****-----"
## [1] "-----*****-----"
## [1] "Without GroupID"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29  4.028  0.1389
##
## Error: id:SHOW_TRUST
##              Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST   1  0.4955  0.4955  11.05 0.00242 **
## Residuals  29  1.3008  0.0449
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##              Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_ID       1  0.4132  0.4132  20.52 9.36e-05 ***
## Residuals  29  0.5840  0.0201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1 0.3741   0.3741   19.36 0.000134 ***
## Residuals          29 0.5603   0.0193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
## [1] "ANOVA analysis on arcsin transformation of send proportion"
## [1] "Without GroupID"
##
## Error: id
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Residuals 29  7.572   0.2611
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST  1 0.9225   0.9225   10.3 0.00323 **
## Residuals 29 2.5962   0.0895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID     1 0.5761   0.5761   15.45 0.000483 ***
## Residuals 29 1.0815   0.0373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1 0.5119   0.5119   16.39 0.000351 ***
## Residuals          29 0.9058   0.0312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##           SS num Df Error SS den Df      F
## (Intercept)      35.309      1  3.4882    25 253.0673
## GroupID           4.084      4  3.4882    25  7.3177
## SHOW_TRUST        0.922      1  1.5686    25  3.5906
## GroupID:SHOW_TRUST 1.028      4  1.5686    25  4.0946
## SHOW_ID           0.576      1  0.8915    25 12.1284
## GroupID:SHOW_ID    0.190      4  0.8915    25  1.3320
## SHOW_TRUST:SHOW_ID 0.512      1  0.6830    25  9.1919
## GroupID:SHOW_TRUST:SHOW_ID 0.223      4  0.6830    25  2.0383
##
##           Pr(>F)
## (Intercept) 1.38e-14 ***
## GroupID     0.0004798 ***
## SHOW_TRUST  0.1310083
## GroupID:SHOW_TRUST 0.0109383 *
## SHOW_ID     0.0252926 *

```

```

## GroupID:SHOW_ID          0.2856253
## SHOW_TRUST:SHOW_ID      0.0387114 *
## GroupID:SHOW_TRUST:SHOW_ID 0.1195922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "ANOVA analysis with absolute profit on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)    270.497     1   73.067    25 92.5503
## GroupID         29.167     4   73.067    25  2.4949
## SHOW_TRUST      95.305     1   49.426    25 19.6997
## GroupID:SHOW_TRUST 19.351     4   49.426    25  2.4470
## SHOW_ID        59.365     1   43.811    25 15.0288
## GroupID:SHOW_ID  15.800     4   43.811    25  2.2541
## SHOW_TRUST:SHOW_ID 43.778     1   30.744    25 60.8886
## GroupID:SHOW_TRUST:SHOW_ID  2.876     4   30.744    25  0.5847
##
##              Pr(>F)
## (Intercept)    6.962e-10 ***
## GroupID        0.068572 .
## SHOW_TRUST     0.011348 *
## GroupID:SHOW_TRUST 0.072647 .
## SHOW_ID       0.017891 *
## GroupID:SHOW_ID  0.091828 .
## SHOW_TRUST:SHOW_ID 0.001455 **
## GroupID:SHOW_TRUST:SHOW_ID 0.676646
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "-----*-----"
## [1] "-----*-----"
## [1] "Without GroupID"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29  102.2    3.525
##
## Error: id:SHOW_TRUST
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST   1   95.30   95.30  40.19 6.31e-07 ***
## Residuals  29   68.78    2.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID      1   59.37   59.37  28.88 8.97e-06 ***
## Residuals  29   59.61    2.06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID

```

```

##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1  43.78   43.78   37.76 1.07e-06 ***
## Residuals          29  33.62    1.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
## [1] "ANOVA analysis with absolute sending amount on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##              SS num Df Error SS den Df      F
## (Intercept)      2787.96      1  202.363    25 344.4249
## GroupID           200.40      4  202.363    25   6.1893
## SHOW_TRUST         49.55      1   91.768    25   5.1728
## GroupID:SHOW_TRUST  38.32      4   91.768    25   2.6095
## SHOW_ID            41.32      1   48.582    25  16.8279
## GroupID:SHOW_ID      9.82      4   48.582    25   1.2634
## SHOW_TRUST:SHOW_ID  37.41      1   39.138    25   8.8590
## GroupID:SHOW_TRUST:SHOW_ID 16.89      4   39.138    25   2.6977
##              Pr(>F)
## (Intercept)      3.916e-16 ***
## GroupID           0.001325 **
## SHOW_TRUST        0.085315 .
## GroupID:SHOW_TRUST 0.059743 .
## SHOW_ID            0.014827 *
## GroupID:SHOW_ID     0.310568
## SHOW_TRUST:SHOW_ID  0.040884 *
## GroupID:SHOW_TRUST:SHOW_ID 0.053764 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "-----"
## [1] "-----"
## [1] "Without GroupID"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29  402.8   13.89
##
## Error: id:SHOW_TRUST
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST  1  49.55   49.55   11.05 0.00242 **
## Residuals  29 130.08    4.49
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID      1  41.32   41.32   20.52 9.36e-05 ***
## Residuals  29  58.40    2.01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1  37.41    37.41   19.36 0.000134 ***
## Residuals          29   56.03     1.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
## [1] "ANOVA analysis with relative sending on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##           SS num Df Error SS den Df      F
## (Intercept)      20.1800      1  2.11849    25 238.1416
## GroupID           0.1004      4  2.11849    25  0.2961
## SHOW_TRUST        0.2764      1  0.60535    25 74.4387
## GroupID:SHOW_TRUST 0.0149      4  0.60535    25  0.1533
## SHOW_ID           0.2448      1  0.30878    25 35.8616
## GroupID:SHOW_ID    0.0273      4  0.30878    25  0.5526
## SHOW_TRUST:SHOW_ID 0.1513      1  0.21871    25  6.9656
## GroupID:SHOW_TRUST:SHOW_ID 0.0869      4  0.21871    25  2.4836
##           Pr(>F)
## (Intercept)      2.757e-14 ***
## GroupID           0.8777170
## SHOW_TRUST        0.0009923 ***
## GroupID:SHOW_TRUST 0.9597047
## SHOW_ID           0.0039100 **
## GroupID:SHOW_ID    0.6989363
## SHOW_TRUST:SHOW_ID 0.0576258 .
## GroupID:SHOW_TRUST:SHOW_ID 0.0695118 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "-----"
## [1] "-----"
## [1] "Without GroupID"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29  2.219 0.07651
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST  1 0.2764 0.27640   12.92 0.00119 **
## Residuals  29 0.6202 0.02139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID    1 0.2448 0.24476   21.12 7.79e-05 ***
## Residuals  29 0.3361 0.01159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1 0.1514 0.15135    14.36 0.000706 ***
## Residuals          29 0.3056 0.01054
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
## [1] "ANOVA analysis on arcsin transformation of send proportion"
## [1] "Without GroupID"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 29  2.717 0.09369
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST  1  0.316 0.31605    12.5 0.00139 **
## Residuals  29  0.733 0.02528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID    1 0.2766  0.2766    20.49 9.43e-05 ***
## Residuals 29 0.3914  0.0135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1 0.1766 0.17659    14.74 0.000617 ***
## Residuals          29 0.3474 0.01198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##           SS num Df Error SS den Df      F
## (Intercept)      22.3170      1 2.58573    25 215.7706
## GroupID           0.1313      4 2.58573    25  0.3175
## SHOW_TRUST        0.3160      1 0.71785    25 83.5416
## GroupID:SHOW_TRUST 0.0151      4 0.71785    25  0.1318
## SHOW_ID           0.2766      1 0.35903    25 34.1342
## GroupID:SHOW_ID    0.0324      4 0.35903    25  0.5643
## SHOW_TRUST:SHOW_ID 0.1766      1 0.24621    25  6.9829
## GroupID:SHOW_TRUST:SHOW_ID 0.1012      4 0.24621    25  2.5678
##
##           Pr(>F)
## (Intercept)      8.41e-14 ***
## GroupID          0.8635589
## SHOW_TRUST       0.0007952 ***
## GroupID:SHOW_TRUST 0.9692902
## SHOW_ID          0.0042793 **

```

```

## GroupID:SHOW_ID          0.6907771
## SHOW_TRUST:SHOW_ID       0.0574293 .
## GroupID:SHOW_TRUST:SHOW_ID 0.0628111 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "ANOVA analysis with absolute profit on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##              SS num Df Error SS den Df      F
## (Intercept)    7724.6      1   343.58    25 562.0756
## GroupID         676.7      4   343.58    25 12.3106
## SHOW_TRUST      13.7      1   123.72    25  0.5562
## GroupID:SHOW_TRUST 98.8      4   123.72    25  4.9887
## SHOW_ID         26.0      1    81.46    25 17.3666
## GroupID:SHOW_ID   6.0      4    81.46    25  0.4600
## SHOW_TRUST:SHOW_ID 28.7      1    72.76    25  1.2332
## GroupID:SHOW_TRUST:SHOW_ID 93.0      4    72.76    25  7.9874
##
##              Pr(>F)
## (Intercept)    < 2.2e-16 ***
## GroupID        1.146e-05 ***
## SHOW_TRUST     0.4972433
## GroupID:SHOW_TRUST 0.0042726 **
## SHOW_ID        0.0140616 *
## GroupID:SHOW_ID  0.7642827
## SHOW_TRUST:SHOW_ID 0.3290364
## GroupID:SHOW_TRUST:SHOW_ID 0.0002719 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "-----*-----"
## [1] "-----*-----"
## [1] "Without GroupID"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29   1020   35.18
##
## Error: id:SHOW_TRUST
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST   1   13.73   13.731    1.79  0.191
## Residuals  29  222.47    7.671
##
## Error: id:SHOW_ID
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID      1   26.03   26.030    8.632 0.00641 **
## Residuals  29   87.45    3.016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1   28.67   28.668    5.016 0.0329 *

```



```

## Residuals          29 165.74   5.715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
## [1] "ANOVA analysis with absolute sending amount on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "With GroupID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##              SS num Df Error SS den Df          F    Pr(>F)
## (Intercept)      4554.0      1   684.95     25 166.2142 1.52e-12
## GroupID          402.5      4   684.95     25   3.6730 0.017407
## SHOW_TRUST       230.7      1   161.97     25   9.1017 0.039283
## GroupID:SHOW_TRUST 101.4      4   161.97     25   3.9115 0.013360
## SHOW_ID          211.8      1   116.00     25  35.7505 0.003932
## GroupID:SHOW_ID    23.7      4   116.00     25   1.2769 0.305509
## SHOW_TRUST:SHOW_ID 132.9      1    58.78     25  13.5700 0.021134
## GroupID:SHOW_TRUST:SHOW_ID 39.2      4    58.78     25   4.1643 0.010143
##
## (Intercept)          ***
## GroupID              *
## SHOW_TRUST           *
## GroupID:SHOW_TRUST   *
## SHOW_ID              **
## GroupID:SHOW_ID
## SHOW_TRUST:SHOW_ID   *
## GroupID:SHOW_TRUST:SHOW_ID *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "-----"
## [1] "-----"
## [1] "Without GroupID"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  29   1088    37.5
##
## Error: id:SHOW_TRUST
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST   1   230.7   230.65    25.4 2.27e-05 ***
## Residuals  29   263.3    9.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##              Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID      1   211.8   211.82   43.97 2.87e-07 ***
## Residuals  29   139.7    4.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##              Df Sum Sq Mean Sq F value    Pr(>F)

```

```
## SHOW_TRUST:SHOW_ID  1 132.87  132.87   39.34 7.57e-07 ***
## Residuals           29 97.95    3.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "-----"
```

Summary by information showed to users

##	Row.names	RelSender	RelReceiver	ProfitSender	ProfitReceiver
## 1	Without Trust	0.4153333	0.3512745	0.576	7.730667
## 2	With Trust	0.5370667	0.4762693	2.384	8.357333
## 3	Without ID	0.4145333	0.3688137	0.748	7.542667
## 4	With ID	0.5378667	0.4587301	2.212	8.545333

```
## Group effect on game_setting
```

Date:Sat 10-Jun-2017 Here, we create a new variable called game\_setting which runs from 1 to 4 correspond to 4 games.

```
## [1] "ANOVA analysis with sending proportion on GroupID:game_setting."
## [1] "Type:  SENDER"
## [1] "-----*****-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df          F      Pr(>F)
## (Intercept)    27.8796      1   2.0236      25 344.4249 3.916e-16 ***
## GroupID         2.0040      4   2.0236      25   6.1893 0.001325 **
## game_setting     1.2828      3   1.7949      75   7.8905 0.003584 **
## GroupID:game_setting 0.6503     12   1.7949      75   2.2644 0.016456 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
## [1] "ANOVA analysis with sending proportion on GroupID:game_setting."
## [1] "Type:  RECEIVER"
## [1] "-----*****-----"
## [1] "ANOVA 2-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##              SS num Df Error SS den Df          F      Pr(>F)
## (Intercept)    20.1800      1   2.1185      25 238.1416 2.757e-14 ***
## GroupID         0.1004      4   2.1185      25   0.2961   0.8777
## game_setting     0.6725      3   1.1328      75  20.8428 4.744e-05 ***
## GroupID:game_setting 0.1291     12   1.1328      75   0.7121   0.7351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----"
```

## Analyze the effect on each group

We want to see effect of TRUST and ID on each group (above is for all groups). Because of the less power, we cannot expect the same significant level as whole data, but the effect are similar between groups.

Because there is only group interaction on Profit of GroupID:SHOW\_TRUST:SHOW\_ID, so we only analyze the Profit of each group.

```
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  1  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  14.18   2.835
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  19.00  18.998   3.995  0.102
## Residuals   5   23.78   4.756
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID     1   5.396   5.396   2.555  0.171
## Residuals   5  10.562   2.112
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1   6.917   6.917   6.37 0.0529 .
## Residuals           5   5.429   1.086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Without Trust With Trust
## 1 Without ID    -0.6150072   2.238112
## 2   With ID     1.4070166   2.112747
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  2  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  28.47   5.694
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  1.6473   1.6473  11.62 0.0191 *
## Residuals   5  0.7085   0.1417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID     1  32.95  32.95  15.38 0.0112 *
## Residuals   5  10.71   2.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
```

```

## SHOW_TRUST:SHOW_ID  1 13.236  13.236   11.38 0.0198 *
## Residuals          5  5.816   1.163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Without Trust With Trust
## 1 Without ID    -0.7637529   1.245500
## 2   With ID     3.0649558   2.103641
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  3  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  17.37   3.473
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST  1  54.97   54.97   33.86 0.00212 **
## Residuals   5   8.12    1.62
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_ID    1  27.465  27.465   33.87 0.00212 **
## Residuals   5   4.055   0.811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  16.55  16.550   6.452 0.0519 .
## Residuals          5  12.82   2.565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           Without Trust With Trust
## 1 Without ID    -1.526897   3.160656
## 2   With ID     2.273431   3.639355
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  4  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5   6.738   1.348
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  14.886  14.886   11.74 0.0187 *
## Residuals   5   6.341   1.268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_ID    1   8.245   8.245   18.64 0.00759 **
## Residuals   5   2.212   0.442
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  4.641    4.641    6.111 0.0564 .
## Residuals          5  3.797    0.759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Without Trust With Trust
## 1 Without ID      0.3579004    2.812525
## 2   With ID      2.4096348    3.105261
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  5  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  6.317    1.263
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  24.16   24.158   11.52 0.0194 *
## Residuals   5  10.48    2.096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1   1.11    1.110    0.341  0.585
## Residuals  5  16.27    3.254
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  5.310    5.310    9.228 0.0288 *
## Residuals          5  2.877    0.575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Without Trust With Trust
## 1 Without ID     -0.9380342    2.009236
## 2   With ID      0.4327228    1.498575
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  1  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  58.13   11.63
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  0.287    0.2872    0.139  0.724
## Residuals   5 10.300    2.0601
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1  11.50   11.502    3.364  0.126
## Residuals  5  17.09    3.419
##

```

```

## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  54.05   54.05   15.29 0.0113 *
## Residuals          5  17.67    3.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  2  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  642.8   128.6
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  145.3   145.28   9.336 0.0282 *
## Residuals   5   77.8    15.56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_ID    1 114.10   114.10   21.25 0.00579 **
## Residuals  5  26.84    5.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  19.77   19.77   0.965  0.371
## Residuals          5 102.38   20.48
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  3  for type:  SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5   40.6    8.121
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1   4.039   4.039   1.544  0.269
## Residuals   5 13.082   2.616
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1  27.33   27.329   10.38 0.0234 *
## Residuals  5  13.17    2.634
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1 13.021   13.021   11.6 0.0191 *
## Residuals          5   5.615    1.123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group: 4 for type: SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  259.2    51.83
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1   57.18    57.18   3.354  0.127
## Residuals   5   85.24    17.05
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID     1    0.02    0.015   0.001  0.973
## Residuals   5   60.78   12.156
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  36.62   36.62   3.308  0.129
## Residuals           5  55.35   11.07
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group: 5 for type: SENDER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  126.5    25.3
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  167.4   167.37   18.15 0.00801 **
## Residuals   5   46.1    9.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID     1 296.23   296.23   24.84 0.00416 **
## Residuals   5   59.64   11.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1 171.35   171.3   25.18 0.00404 **
## Residuals           5   34.02    6.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group: 1 for type: RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5   46.6    9.32
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)

```



```

## SHOW_TRUST 1 1.09 1.092 0.123 0.74
## Residuals 5 44.36 8.872
##
## Error: id:SHOW_ID
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID 1 10.33 10.332 2.119 0.205
## Residuals 5 24.38 4.876
##
## Error: id:SHOW_TRUST:SHOW_ID
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1 97.34 97.34 30.93 0.00258 **
## Residuals 5 15.73 3.15
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Without Trust With Trust
## 1 Without ID 2.754401 7.20873
## 2 With ID 8.094452 4.49323
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group: 2 for type: RECEIVER"
##
## Error: id
## Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 5 36.06 7.212
##
## Error: id:SHOW_TRUST
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 2.793 2.793 1.045 0.354
## Residuals 5 13.363 2.673
##
## Error: id:SHOW_ID
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID 1 7.637 7.637 1.791 0.238
## Residuals 5 21.316 4.263
##
## Error: id:SHOW_TRUST:SHOW_ID
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1 0.5754 0.5754 1.199 0.323
## Residuals 5 2.3990 0.4798
## Without Trust With Trust
## 1 Without ID 4.763095 5.755056
## 2 With ID 6.200974 6.573557
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group: 3 for type: RECEIVER"
##
## Error: id
## Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 5 210.2 42.05
##
## Error: id:SHOW_TRUST
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 92.87 92.87 50.59 0.000852 ***
## Residuals 5 9.18 1.84
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID

```

```

##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1  6.542    6.542    1.319  0.303
## Residuals   5 24.793    4.959
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  1.546    1.546    0.315  0.599
## Residuals           5 24.529    4.906
##           Without Trust With Trust
## 1 Without ID        9.349393   13.79120
## 2   With ID       10.901195   14.32778
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  4  for type:  RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals   5  12.01    2.401
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1   0.01    0.010    0.003  0.96
## Residuals   5  18.36    3.672
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1  7.496    7.496    6.631 0.0497 *
## Residuals   5  5.652    1.130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  11.44   11.444    4.092  0.099 .
## Residuals           5  13.98    2.797
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Without Trust With Trust
## 1 Without ID        7.751158    9.090500
## 2   With ID       10.250000    8.827165
## [1] "Analyze Profit of SHOW_TRUST and SHOW_ID interaction on Group:  5  for type:  RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals   5  38.67    7.734
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  15.72   15.719    2.044  0.212
## Residuals   5  38.46    7.692
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID    1   0.018    0.0182    0.017  0.901
## Residuals   5   5.314    1.0628
##

```

```

## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1 10.75 10.749 3.335 0.127
## Residuals      5 16.12 3.223
##           Without Trust With Trust
## 1 Without ID      9.034127 6.077068
## 2 With ID         7.750722 7.470557
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group: 1 for type: RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 5 47.49 9.498
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 3.589 3.589 1.68 0.252
## Residuals 5 10.679 2.136
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID 1 45.46 45.46 5.661 0.0632 .
## Residuals 5 40.15 8.03
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1 152.80 152.80 14.53 0.0125 *
## Residuals      5 52.59 10.52
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group: 2 for type: RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 5 777.3 155.5
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 731.2 731.2 55.27 0.000694 ***
## Residuals 5 66.2 13.2
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID 1 77.27 77.27 12.17 0.0175 *
## Residuals 5 31.75 6.35
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID 1 324.6 324.6 64.47 0.000485 ***

```

```

## Residuals          5    25.2    5.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  3  for type:  RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  67.43   13.49
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  4.191    4.191   2.488  0.176
## Residuals   5  8.421    1.684
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_ID     1  81.37   81.37   17.68 0.00845 **
## Residuals   5  23.02    4.60
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST:SHOW_ID  1  50.31   50.31   55.88 0.000677 ***
## Residuals          5   4.50    0.90
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  4  for type:  RECEIVER"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  380.9   76.19
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST  1 295.82  295.82   83.05 0.000266 ***
## Residuals   5  17.81    3.56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID     1   0.21    0.21   0.005  0.947
## Residuals   5 211.87   42.37
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  113.7  113.71   4.913 0.0775 .
## Residuals          5  115.7   23.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze response time of SHOW_TRUST and SHOW_ID interaction on Group:  5  for type:  RECEIVER"
##
## Error: id

```

```

##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  154.8   30.96
##
## Error: id:SHOW_TRUST
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST  1 262.90  262.90   80.91 0.000283 ***
## Residuals   5   16.25    3.25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_ID     1  529.2   529.2   70.55 0.000392 ***
## Residuals   5   37.5    7.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST:SHOW_ID
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID  1  517.9   517.9   41.17 0.00136 **
## Residuals           5   62.9   12.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

## Test the regression of sending behavior on different pieces of information

In the above section, we consider all games in a same kind as a whole. In this section, we want to see how each person behave in different kinds of situation.

First, we define four empty data frames for 4 kinds of game.

The new variable 'peak\_end\_trust' is used for calculate "peak end effect", which basically said that the feeling about a repeated event is average of maximum feeling so far and the last feeling the subject have with this event.

Then again we run through the whole data and load the game to corresponding dataframes.

After that, we applied the analysis on each individual game.

```
## [1] "Linear regression of relative sending on trust value of Simple Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16900 -0.03798 -0.02209  0.04528  0.18161
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.2611     0.0614  -4.253 0.000226 ***
## trust_value     0.2562     0.1554   1.649 0.110772
## my_trust_value  1.4889     0.1163  12.801 5.57e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08502 on 27 degrees of freedom
## Multiple R-squared:  0.8644, Adjusted R-squared:  0.8543
## F-statistic: 86.03 on 2 and 27 DF,  p-value: 1.936e-12
##
## [1] "Linear regression of relative sending on trust value of ID Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17997 -0.08236  0.01791  0.04621  0.24022
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.3679     0.1688  -2.179  0.0382 *
## trust_value     0.5488     0.3173   1.730  0.0951 .
## my_trust_value  1.4517     0.1559   9.311 6.41e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1014 on 27 degrees of freedom
## Multiple R-squared:  0.7634, Adjusted R-squared:  0.7459
## F-statistic: 43.57 on 2 and 27 DF,  p-value: 3.533e-09
```

```
##
## [1] "Linear regression of relative sending on trust value of Score Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.210241 -0.056944  0.009976  0.061286  0.128565
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.6727     0.0893  -7.533 4.20e-08 ***
## trust_value     1.3192     0.2320   5.686 4.85e-06 ***
## my_trust_value  1.3565     0.1843   7.360 6.45e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08596 on 27 degrees of freedom
## Multiple R-squared:  0.8879, Adjusted R-squared:  0.8796
## F-statistic: 106.9 on 2 and 27 DF,  p-value: 1.481e-13
##
## [1] "Linear regression of relative sending on trust value of Combine Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.150939 -0.044539  0.002393  0.038702  0.198586
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.6266     0.0851  -7.364 6.39e-08 ***
## trust_value     1.0693     0.2279   4.692 6.96e-05 ***
## my_trust_value  1.4954     0.1795   8.332 6.10e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08923 on 27 degrees of freedom
## Multiple R-squared:  0.8966, Adjusted R-squared:  0.889
## F-statistic: 117.1 on 2 and 27 DF,  p-value: 4.938e-14
##
## [1] "Linear regression of relative sending on trust value of Simple Game for type:  RECEIVER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21929 -0.10059  0.00804  0.06272  0.39919
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept)      0.2112      0.1187    1.780    0.0864 .
## trust_value     -0.8357      0.3103   -2.693    0.0120 *
## my_trust_value    1.0568      0.1676    6.304 9.54e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1354 on 27 degrees of freedom
## Multiple R-squared:  0.6397, Adjusted R-squared:  0.613
## F-statistic: 23.97 on 2 and 27 DF,  p-value: 1.036e-06
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value + AbsPartnerSend,
##     data = df_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20487 -0.09520 -0.00004  0.07722  0.38307
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.11234    0.15590   0.721   0.478
## trust_value   -0.23317    0.68953  -0.338   0.738
## my_trust_value  1.11924    0.17951   6.235 1.35e-06 ***
## AbsPartnerSend -0.03873    0.03957  -0.979   0.337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1355 on 26 degrees of freedom
## Multiple R-squared:  0.6525, Adjusted R-squared:  0.6124
## F-statistic: 16.27 on 3 and 26 DF,  p-value: 3.714e-06
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + AbsPartnerSend, data = df_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19912 -0.10103 -0.00098  0.07989  0.38361
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.06621    0.07424   0.892  0.38036
## my_trust_value  1.13920    0.16673   6.833 2.44e-07 ***
## AbsPartnerSend -0.05067    0.01752  -2.892  0.00748 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1333 on 27 degrees of freedom
## Multiple R-squared:  0.6509, Adjusted R-squared:  0.6251
## F-statistic: 25.18 on 2 and 27 DF,  p-value: 6.748e-07
##

```



```

## [1] "Linear regression of relative sending on trust value of ID Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20288 -0.05319 -0.01134  0.06952  0.21452
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.078318   0.157743  -0.496   0.624
## trust_value   -0.002079   0.323062  -0.006   0.995
## my_trust_value  1.145081   0.138444   8.271 7.04e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09207 on 27 degrees of freedom
## Multiple R-squared:  0.717, Adjusted R-squared:  0.6961
## F-statistic: 34.21 on 2 and 27 DF, p-value: 3.969e-08
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value + AbsPartnerSend,
##     data = df_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17328 -0.05515 -0.01478  0.04954  0.20256
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.19742   0.16151  -1.222   0.2325
## trust_value    0.52466   0.40635   1.291   0.2080
## my_trust_value  1.29730   0.15239   8.513 5.41e-09 ***
## AbsPartnerSend -0.03517   0.01778  -1.978   0.0586 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08747 on 26 degrees of freedom
## Multiple R-squared:  0.754, Adjusted R-squared:  0.7257
## F-statistic: 26.57 on 3 and 26 DF, p-value: 4.433e-08
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + AbsPartnerSend, data = df_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18040 -0.06471 -0.01187  0.05895  0.20211
##
## Coefficients:

```

```

##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.01343    0.07695  -0.175    0.863
## my_trust_value  1.23312    0.14582   8.456 4.55e-09 ***
## AbsPartnerSend -0.02012    0.01359  -1.480    0.150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08855 on 27 degrees of freedom
## Multiple R-squared:  0.7383, Adjusted R-squared:  0.7189
## F-statistic: 38.08 on 2 and 27 DF,  p-value: 1.384e-08
##
## [1] "Linear regression of relative sending on trust value of Score Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.280109 -0.051707  0.002227  0.056455  0.201905
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.2574    0.1374   1.873   0.0720 .
## trust_value    -0.7084    0.3553  -1.994   0.0564 .
## my_trust_value  1.1133    0.1968   5.656 5.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1149 on 27 degrees of freedom
## Multiple R-squared:  0.5491, Adjusted R-squared:  0.5157
## F-statistic: 16.44 on 2 and 27 DF,  p-value: 2.137e-05
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value + AbsPartnerSend,
##     data = df_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.251584 -0.060449 -0.008884  0.054348  0.239598
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.01454    0.20506   0.071   0.944
## trust_value    -0.07442    0.53315  -0.140   0.890
## my_trust_value  1.49678    0.31135   4.807 5.59e-05 ***
## AbsPartnerSend -0.03995    0.02555  -1.564   0.130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1119 on 26 degrees of freedom
## Multiple R-squared:  0.5879, Adjusted R-squared:  0.5403
## F-statistic: 12.36 on 3 and 26 DF,  p-value: 3.252e-05

```

```

##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + AbsPartnerSend, data = df_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.250120 -0.061284 -0.008514  0.055041  0.240287
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.01182    0.07835   -0.151   0.8812
## my_trust_value  1.51418    0.28009   5.406 1.02e-05 ***
## AbsPartnerSend -0.04267    0.01629  -2.619   0.0143 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1099 on 27 degrees of freedom
## Multiple R-squared:  0.5876, Adjusted R-squared:  0.557
## F-statistic: 19.23 on 2 and 27 DF,  p-value: 6.415e-06
##
## [1] "Linear regression of relative sending on trust value of Combine Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value, data = df_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16794 -0.05460 -0.02329  0.07147  0.17540
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.40295    0.09903   4.069 0.000369 ***
## trust_value   -0.70284    0.28055  -2.505 0.018572 *
## my_trust_value  0.82204    0.16628   4.944 3.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09873 on 27 degrees of freedom
## Multiple R-squared:  0.4815, Adjusted R-squared:  0.4431
## F-statistic: 12.54 on 2 and 27 DF,  p-value: 0.000141
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ trust_value + my_trust_value + AbsPartnerSend,
##     data = df_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18671 -0.06514 -0.01169  0.07493  0.15791
##
## Coefficients:

```

```

##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.30593    0.18356   1.667 0.107599
## trust_value   -0.37417    0.59335  -0.631 0.533803
## my_trust_value  0.91062    0.21910   4.156 0.000311 ***
## AbsPartnerSend -0.01687    0.02674  -0.631 0.533744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09985 on 26 degrees of freedom
## Multiple R-squared:  0.4893, Adjusted R-squared:  0.4304
## F-statistic: 8.303 on 3 and 26 DF,  p-value: 0.0004876
##
## [1] "-----"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + AbsPartnerSend, data = df_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.204412 -0.063405  0.007548  0.073628  0.155424
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.19722    0.06236   3.162 0.00384 **
## my_trust_value  0.95641    0.20441   4.679 7.22e-05 ***
## AbsPartnerSend -0.03168    0.01265  -2.505 0.01857 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09873 on 27 degrees of freedom
## Multiple R-squared:  0.4815, Adjusted R-squared:  0.4431
## F-statistic: 12.54 on 2 and 27 DF,  p-value: 0.000141

```

## Group effect analysis with group and round number as between variables

Date: 2-Jun-2017

```
## [1] "Analyze SENDER"
##
## Error: id
##


|                                 | Df | Sum Sq | Mean Sq | F     | value | Pr(>F) |
|---------------------------------|----|--------|---------|-------|-------|--------|
| SHOW_TRUST                      | 1  | 1.40   | 1.395   | 0.395 | 0.544 |        |
| SHOW_ID                         | 1  | 0.50   | 0.505   | 0.143 | 0.713 |        |
| round_number                    | 4  | 5.75   | 1.437   | 0.407 | 0.800 |        |
| SHOW_TRUST:SHOW_ID              | 1  | 0.10   | 0.101   | 0.029 | 0.869 |        |
| SHOW_TRUST:round_number         | 4  | 1.47   | 0.368   | 0.104 | 0.978 |        |
| SHOW_ID:round_number            | 4  | 0.76   | 0.190   | 0.054 | 0.994 |        |
| SHOW_TRUST:SHOW_ID:round_number | 4  | 7.52   | 1.879   | 0.532 | 0.716 |        |
| Residuals                       | 10 | 35.33  | 3.533   |       |       |        |


##
## Error: SHOW_ID
##


|            | Df | Sum Sq | Mean Sq |
|------------|----|--------|---------|
| SHOW_TRUST | 1  | 5.361  | 5.361   |


##
## Error: round_number
##


|              | Df | Sum Sq | Mean Sq |
|--------------|----|--------|---------|
| SHOW_TRUST   | 1  | 0.3649 | 0.3649  |
| round_number | 3  | 0.5964 | 0.1988  |


##
## Error: id:SHOW_ID
##


|                                 | Df | Sum Sq | Mean Sq | F     | value | Pr(>F) |
|---------------------------------|----|--------|---------|-------|-------|--------|
| SHOW_TRUST                      | 1  | 0.359  | 0.3590  | 1.324 | 0.268 |        |
| SHOW_TRUST:SHOW_ID              | 1  | 0.052  | 0.0523  | 0.193 | 0.667 |        |
| SHOW_TRUST:round_number         | 4  | 0.727  | 0.1817  | 0.670 | 0.623 |        |
| SHOW_ID:round_number            | 4  | 0.217  | 0.0542  | 0.200 | 0.935 |        |
| SHOW_TRUST:SHOW_ID:round_number | 4  | 1.903  | 0.4757  | 1.754 | 0.191 |        |
| Residuals                       | 15 | 4.068  | 0.2712  |       |       |        |


##
## Error: id:round_number
##


|                                 | Df  | Sum Sq | Mean Sq | F     | value  | Pr(>F) |
|---------------------------------|-----|--------|---------|-------|--------|--------|
| SHOW_TRUST                      | 1   | 0.080  | 0.0803  | 0.808 | 0.3708 |        |
| SHOW_TRUST:SHOW_ID              | 1   | 0.440  | 0.4400  | 4.430 | 0.0378 | *      |
| SHOW_TRUST:round_number         | 4   | 0.264  | 0.0660  | 0.665 | 0.6180 |        |
| SHOW_ID:round_number            | 4   | 0.670  | 0.1675  | 1.687 | 0.1588 |        |
| SHOW_TRUST:SHOW_ID:round_number | 4   | 0.101  | 0.0253  | 0.254 | 0.9064 |        |
| Residuals                       | 102 | 10.129 | 0.0993  |       |        |        |


## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: SHOW_ID:round_number
##


|                         | Df | Sum Sq | Mean Sq |
|-------------------------|----|--------|---------|
| SHOW_TRUST              | 1  | 0.0886 | 0.0886  |
| SHOW_TRUST:SHOW_ID      | 1  | 0.0275 | 0.0275  |
| SHOW_TRUST:round_number | 2  | 0.6357 | 0.3178  |


##
## Error: id:GroupID:SHOW_TRUST
##


|  | Df | Sum Sq | Mean Sq | F | value | Pr(>F) |
|--|----|--------|---------|---|-------|--------|
|--|----|--------|---------|---|-------|--------|


```

```

## SHOW_TRUST          1  5.841    5.841    9.329 0.00626 **
## SHOW_TRUST:SHOW_ID  1  0.004    0.004    0.006 0.93990
## SHOW_TRUST:round_number  4  2.283    0.571    0.911 0.47634
## SHOW_TRUST:SHOW_ID:round_number  4  0.217    0.054    0.087 0.98562
## Residuals          20 12.523    0.626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID:round_number
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID      1  0.003  0.00322    0.055  0.815
## SHOW_TRUST:round_number  4  0.205  0.05126    0.876  0.481
## SHOW_TRUST:SHOW_ID:round_number  4  0.073  0.01824    0.311  0.870
## Residuals             107  6.264  0.05854
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID      1  4.094   4.094 16.809 0.000512 ***
## SHOW_TRUST:round_number  4  0.296   0.074   0.304 0.871920
## SHOW_TRUST:SHOW_ID:round_number  4  0.917   0.229   0.942 0.459361
## Residuals             21  5.115   0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:round_number
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:round_number  4  1.227  0.30663   4.384 0.00249 **
## SHOW_TRUST:SHOW_ID:round_number  4  0.127  0.03171   0.453 0.76968
## Residuals             112  7.833  0.06994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID:round_number
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID:round_number  4  0.731  0.18265   1.846  0.126
## Residuals             101  9.993  0.09894
##
## Error: Within
##
## Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 915  58.76  0.06422
## [1] "Analyze RECEIVER"
##
## Error: id
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST          1  6.192   6.192 12.341 0.0056 **
## SHOW_ID              1  0.058   0.058   0.116 0.7402
## round_number         4  2.178   0.545   1.085 0.4144
## SHOW_TRUST:SHOW_ID    1  0.731   0.731   1.457 0.2552
## SHOW_TRUST:round_number  4  0.813   0.203   0.405 0.8012
## SHOW_ID:round_number   4  2.950   0.738   1.470 0.2824
## SHOW_TRUST:SHOW_ID:round_number  4  3.935   0.984   1.961 0.1768
## Residuals           10  5.018   0.502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

##
## Error: SHOW_ID
##           Df Sum Sq Mean Sq
## SHOW_TRUST 1  1.885    1.885
##
## Error: round_number
##           Df Sum Sq Mean Sq
## SHOW_TRUST  1 0.00122 0.00122
## round_number 3 0.19468 0.06489
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1 0.0852 0.08516   0.602  0.450
## SHOW_TRUST:SHOW_ID  1 0.0001 0.00010   0.001  0.979
## SHOW_TRUST:round_number  4 0.3215 0.08037   0.568  0.690
## SHOW_ID:round_number  4 0.3699 0.09247   0.654  0.633
## SHOW_TRUST:SHOW_ID:round_number  4 0.1621 0.04054   0.287  0.882
## Residuals      15 2.1211 0.14140
##
## Error: id:round_number
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1  0.246 0.24590   7.499 0.00729 **
## SHOW_TRUST:SHOW_ID  1  0.189 0.18875   5.756 0.01825 *
## SHOW_TRUST:round_number  4  0.310 0.07760   2.367 0.05769 .
## SHOW_ID:round_number  4  0.174 0.04362   1.330 0.26385
## SHOW_TRUST:SHOW_ID:round_number  4  0.143 0.03569   1.088 0.36638
## Residuals     102  3.345 0.03279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: SHOW_ID:round_number
##           Df Sum Sq Mean Sq
## SHOW_TRUST  1 0.12037 0.12037
## SHOW_TRUST:SHOW_ID  1 0.00391 0.00391
## SHOW_TRUST:round_number  2 0.00040 0.00020
##
## Error: id:GroupID:SHOW_TRUST
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1 2.4551  2.4551  24.107 8.46e-05 ***
## SHOW_TRUST:SHOW_ID  1 0.0456  0.0456   0.448  0.5109
## SHOW_TRUST:round_number  4 1.2937  0.3234   3.176  0.0357 *
## SHOW_TRUST:SHOW_ID:round_number  4 0.5540  0.1385   1.360  0.2830
## Residuals     20 2.0368  0.1018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID:round_number
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  0.036 0.03598   0.893  0.347
## SHOW_TRUST:round_number  4  0.053 0.01333   0.331  0.857
## SHOW_TRUST:SHOW_ID:round_number  4  0.050 0.01257   0.312  0.869
## Residuals     106  4.272 0.04030
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID

```

```

##                                Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID            1  1.6608   1.6608  22.142 0.00012 ***
## SHOW_TRUST:round_number        4  0.7022   0.1756   2.340 0.08830 .
## SHOW_TRUST:SHOW_ID:round_number 4  0.1656   0.0414   0.552 0.69969
## Residuals                     21  1.5751   0.0750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:round_number
##                                Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:round_number        4  0.033 0.00837   0.202 0.937
## SHOW_TRUST:SHOW_ID:round_number 4  0.140 0.03500   0.846 0.499
## Residuals                     105  4.345 0.04138
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID:round_number
##                                Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID:round_number 4  0.173 0.04325   1.253 0.296
## Residuals                     80  2.763 0.03453
##
## Error: Within
##                                Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 687  19.53 0.02843

```



## Group effect analysis with group and round number as between variables

However, we keep only first 2 rounds in each roles

Date: 12-Jun-2017

```
## [1] "Analyze SENDER"
##
## Error: id
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 0.005 0.005 0.002 0.961
## SHOW_ID 1 3.572 3.572 1.722 0.219
## round_number 4 3.311 0.828 0.399 0.805
## SHOW_TRUST:SHOW_ID 1 1.080 1.080 0.521 0.487
## SHOW_TRUST:round_number 4 2.718 0.679 0.328 0.853
## SHOW_ID:round_number 4 3.974 0.994 0.479 0.751
## SHOW_TRUST:SHOW_ID:round_number 4 2.990 0.748 0.360 0.831
## Residuals 10 20.744 2.074
##
## Error: SHOW_ID
## Df Sum Sq Mean Sq
## SHOW_TRUST 1 3.298 3.298
##
## Error: round_number
## Df Sum Sq Mean Sq
## SHOW_TRUST 1 0.0496 0.0496
## round_number 3 0.9846 0.3282
##
## Error: id:SHOW_ID
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 0.047 0.0471 0.195 0.665
## SHOW_TRUST:SHOW_ID 1 0.062 0.0620 0.256 0.620
## SHOW_TRUST:round_number 4 0.169 0.0423 0.175 0.948
## SHOW_ID:round_number 4 0.239 0.0598 0.247 0.907
## SHOW_TRUST:SHOW_ID:round_number 4 1.500 0.3750 1.550 0.239
## Residuals 15 3.630 0.2420
##
## Error: id:round_number
##
## Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST 1 0.156 0.1559 1.582 0.2115
## SHOW_TRUST:SHOW_ID 1 0.401 0.4009 4.069 0.0464 *
## SHOW_TRUST:round_number 4 0.788 0.1971 2.000 0.1005
## SHOW_ID:round_number 4 0.541 0.1352 1.373 0.2491
## SHOW_TRUST:SHOW_ID:round_number 4 0.326 0.0815 0.828 0.5108
## Residuals 97 9.557 0.0985
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: SHOW_ID:round_number
## Df Sum Sq Mean Sq
## SHOW_TRUST 1 0.0398 0.03975
## SHOW_TRUST:SHOW_ID 1 0.2562 0.25618
## SHOW_TRUST:round_number 2 0.3463 0.17316
##
## Error: id:GroupID:SHOW_TRUST
```

```

##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST      1  2.389   2.3892   5.825 0.0255 *
## SHOW_TRUST:SHOW_ID      1  0.241   0.2412   0.588 0.4522
## SHOW_TRUST:round_number      4  1.049   0.2622   0.639 0.6406
## SHOW_TRUST:SHOW_ID:round_number      4  2.202   0.5505   1.342 0.2890
## Residuals      20  8.203   0.4101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID:round_number
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID      1  0.275   0.27540   4.702 0.0326 *
## SHOW_TRUST:round_number      4  0.507   0.12668   2.163 0.0789 .
## SHOW_TRUST:SHOW_ID:round_number      4  0.266   0.06650   1.135 0.3445
## Residuals      97  5.681   0.05857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID      1  1.877   1.8767   6.640 0.0176 *
## SHOW_TRUST:round_number      4  0.072   0.0179   0.063 0.9920
## SHOW_TRUST:SHOW_ID:round_number      4  0.232   0.0580   0.205 0.9326
## Residuals      21  5.935   0.2826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:round_number
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:round_number      4  0.666   0.16657   2.081 0.0891 .
## SHOW_TRUST:SHOW_ID:round_number      4  0.319   0.07983   0.997 0.4129
## Residuals      97  7.764   0.08005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID:round_number
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID:round_number      4  0.609   0.15237   1.821 0.135
## Residuals      66  5.524   0.08369
##
## Error: Within
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 540  29.27   0.0542
## [1] "Analyze RECEIVER"
##
## Error: id
##              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST      1  4.313   4.313  15.379 0.00286 **
## SHOW_ID          1  0.169   0.169   0.602 0.45588
## round_number      4  1.707   0.427   1.522 0.26840
## SHOW_TRUST:SHOW_ID      1  1.135   1.135   4.045 0.07203 .
## SHOW_TRUST:round_number      4  2.172   0.543   1.936 0.18086
## SHOW_ID:round_number      4  1.166   0.291   1.039 0.43404
## SHOW_TRUST:SHOW_ID:round_number      4  2.866   0.716   2.555 0.10440

```

```

## Residuals                10  2.805   0.280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: SHOW_ID
##           Df Sum Sq Mean Sq
## SHOW_TRUST 1  1.595   1.595
##
## Error: round_number
##           Df Sum Sq Mean Sq
## SHOW_TRUST  1 0.0038 0.00377
## round_number 3 0.3988 0.13294
##
## Error: id:SHOW_ID
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST  1 0.0750 0.07504   0.830  0.377
## SHOW_TRUST:SHOW_ID  1 0.0263 0.02626   0.290  0.598
## SHOW_TRUST:round_number  4 0.4343 0.10859   1.201  0.351
## SHOW_ID:round_number  4 0.3721 0.09301   1.029  0.424
## SHOW_TRUST:SHOW_ID:round_number  4 0.4084 0.10209   1.129  0.380
## Residuals        15 1.3564 0.09043
##
## Error: id:round_number
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST  1  0.103  0.1034   2.434 0.12183
## SHOW_TRUST:SHOW_ID  1  0.422  0.4218  9.930 0.00213 **
## SHOW_TRUST:round_number  4  0.227  0.0567  1.334 0.26233
## SHOW_ID:round_number  4  0.078  0.0194  0.457 0.76709
## SHOW_TRUST:SHOW_ID:round_number  4  0.088  0.0219  0.516 0.72376
## Residuals        102  4.333  0.0425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: SHOW_ID:round_number
##           Df Sum Sq Mean Sq
## SHOW_TRUST  1 0.05270 0.05270
## SHOW_TRUST:SHOW_ID  1 0.00013 0.00013
## SHOW_TRUST:round_number  2 0.03535 0.01768
##
## Error: id:GroupID:SHOW_TRUST
##           Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST  1 1.9292  1.9292 18.362 0.000361 ***
## SHOW_TRUST:SHOW_ID  1 0.0562  0.0562  0.535 0.472847
## SHOW_TRUST:round_number  4 0.3494  0.0874  0.831 0.520943
## SHOW_TRUST:SHOW_ID:round_number  4 0.5875  0.1469  1.398 0.270664
## Residuals        20 2.1013  0.1051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_ID:round_number
##           Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID  1  0.040 0.03987   1.062  0.305
## SHOW_TRUST:round_number  4  0.054 0.01345   0.358  0.838
## SHOW_TRUST:SHOW_ID:round_number  4  0.099 0.02466   0.657  0.623

```

```

## Residuals          91  3.416 0.03753
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID
##               Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID      1  1.2704    1.2704   15.718 0.000707 ***
## SHOW_TRUST:round_number  4  0.1561    0.0390    0.483 0.747990
## SHOW_TRUST:SHOW_ID:round_number  4  0.1240    0.0310    0.384 0.817785
## Residuals             21  1.6973    0.0808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:GroupID:SHOW_TRUST:round_number
##               Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:round_number  4  0.074 0.01862    0.397  0.81
## SHOW_TRUST:SHOW_ID:round_number  4  0.335 0.08368    1.783  0.14
## Residuals             84  3.941 0.04692
##
## Error: id:GroupID:SHOW_TRUST:SHOW_ID:round_number
##               Df Sum Sq Mean Sq F value    Pr(>F)
## SHOW_TRUST:SHOW_ID:round_number  3  0.0982 0.03272    1.408  0.254
## Residuals             42  0.9762 0.02324
##
## Error: Within
##               Df Sum Sq Mean Sq F value    Pr(>F)
## Residuals      440  13.72 0.03119

```

## Group effect analysis for each round

However, we keep only first 2 rounds in each roles

Date: 12-Jun-2017

```
## [1] "Analyze SENDER"
## [1] "Round number: 1"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)      34.992      1   5.7358      25 152.5149
## GroupID           1.777      4   5.7358      25  1.9365
## SHOW_TRUST        0.481      1   1.2942      25  3.7789
## GroupID:SHOW_TRUST 0.510      4   1.2942      25  2.4606
## SHOW_ID           0.108      1   1.3642      25  3.2522
## GroupID:SHOW_ID    0.133      4   1.3642      25  0.6086
## SHOW_TRUST:SHOW_ID 0.012      1   1.1025      25  0.3189
## GroupID:SHOW_TRUST:SHOW_ID 0.151      4   1.1025      25  0.8532
##
##              Pr(>F)
## (Intercept)      3.871e-12 ***
## GroupID           0.13556
## SHOW_TRUST        0.12382
## GroupID:SHOW_TRUST 0.07147 .
## SHOW_ID           0.14566
## GroupID:SHOW_ID    0.66021
## SHOW_TRUST:SHOW_ID 0.60241
## GroupID:SHOW_TRUST:SHOW_ID 0.50527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*****--"
## [1] "Analyze SENDER"
## [1] "Round number: 2"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)      20.7936      1   2.891      20 143.8506
## GroupID           0.8954      4   2.891      20  1.5486
## SHOW_TRUST        0.4900      1   1.273      20  2.9385
## GroupID:SHOW_TRUST 0.6670      4   1.273      20  2.6198
## SHOW_ID           0.1444      1   1.067      20  2.2336
## GroupID:SHOW_ID    0.2586      4   1.067      20  1.2118
## SHOW_TRUST:SHOW_ID 0.2304      1   2.561      20  6.2019
## GroupID:SHOW_TRUST:SHOW_ID 0.1486      4   2.561      20  0.2901
##
##              Pr(>F)
## (Intercept)      1.372e-10 ***
## GroupID           0.22655
## SHOW_TRUST        0.16164
## GroupID:SHOW_TRUST 0.06569 .
## SHOW_ID           0.20935
## GroupID:SHOW_ID    0.33687
## SHOW_TRUST:SHOW_ID 0.06746 .
## GroupID:SHOW_TRUST:SHOW_ID 0.88091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## [1] "----*****--"
## [1] "Analyze SENDER"
## [1] "Round number: 3"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)      19.2080      1  2.0463      15 140.8039
## GroupID           2.0907      4  2.0463      15  3.8316
## SHOW_TRUST        0.1620      1  1.4688      15  4.4922
## GroupID:SHOW_TRUST 0.1442      4  1.4688      15  0.3683
## SHOW_ID           0.5445      1  1.7537      15  2.5422
## GroupID:SHOW_ID    0.8568      4  1.7537      15  1.8320
## SHOW_TRUST:SHOW_ID 0.5445      1  1.2563      15  6.3268
## GroupID:SHOW_TRUST:SHOW_ID 0.3443      4  1.2563      15  1.0276
##
##              Pr(>F)
## (Intercept)      5.039e-09 ***
## GroupID           0.02440 *
## SHOW_TRUST        0.10140
## GroupID:SHOW_TRUST 0.82748
## SHOW_ID           0.18607
## GroupID:SHOW_ID    0.17522
## SHOW_TRUST:SHOW_ID 0.06568 .
## GroupID:SHOW_TRUST:SHOW_ID 0.42493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*****--"
## [1] "Analyze SENDER"
## [1] "Round number: 4"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)      28.8120      1  2.9925      25 240.7018
## GroupID           2.9305      4  2.9925      25  6.1205
## SHOW_TRUST        0.6750      1  2.9558      25  4.0652
## GroupID:SHOW_TRUST 0.6642      4  2.9558      25  1.4044
## SHOW_ID           1.5413      1  2.9142      25 32.5347
## GroupID:SHOW_ID    0.1895      4  2.9142      25  0.4064
## SHOW_TRUST:SHOW_ID 0.2083      1  2.4642      25  2.3641
## GroupID:SHOW_TRUST:SHOW_ID 0.3525      4  2.4642      25  0.8941
##
##              Pr(>F)
## (Intercept)      2.442e-14 ***
## GroupID           0.001413 **
## SHOW_TRUST        0.113984
## GroupID:SHOW_TRUST 0.261388
## SHOW_ID           0.004670 **
## GroupID:SHOW_ID    0.802224
## SHOW_TRUST:SHOW_ID 0.198977
## GroupID:SHOW_TRUST:SHOW_ID 0.482125
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*****--"
## [1] "Analyze SENDER"
## [1] "Round number: 5"
## [1] "-----*****-----"

```

```

## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F
## (Intercept)      31.0083      1  2.7842    25 278.4346
## GroupID          1.3675      4  2.7842    25  3.0698
## SHOW_TRUST       1.5413      1  2.2825    25  6.4480
## GroupID:SHOW_TRUST 0.9562      4  2.2825    25  2.6182
## SHOW_ID          0.2613      1  1.4392    25  0.9960
## GroupID:SHOW_ID    1.0495      4  1.4392    25  4.5578
## SHOW_TRUST:SHOW_ID 1.1603      1  3.9675    25 10.9941
## GroupID:SHOW_TRUST:SHOW_ID 0.4222      4  3.9675    25  0.6650
##
##          Pr(>F)
## (Intercept)      4.616e-15 ***
## GroupID          0.034670 *
## SHOW_TRUST       0.064030 .
## GroupID:SHOW_TRUST 0.059125 .
## SHOW_ID          0.374755
## GroupID:SHOW_ID    0.006671 **
## SHOW_TRUST:SHOW_ID 0.029496 *
## GroupID:SHOW_TRUST:SHOW_ID 0.622180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "Analyze RECEIVER"
## [1] "Round number: 1"
## [1] "-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##          SS num Df Error SS den Df      F
## (Intercept)      17.2622      1  3.4351    23 115.5822
## GroupID          0.0725      4  3.4351    23  0.1214
## SHOW_TRUST       0.4794      1  1.0621    23 12.7083
## GroupID:SHOW_TRUST 0.1509      4  1.0621    23  0.8168
## SHOW_ID          0.3258      1  1.0368    23  7.1514
## GroupID:SHOW_ID    0.1823      4  1.0368    23  1.0107
## SHOW_TRUST:SHOW_ID 0.1880      1  0.6186    23 14.9404
## GroupID:SHOW_TRUST:SHOW_ID 0.0503      4  0.6186    23  0.4680
##
##          Pr(>F)
## (Intercept)      1.917e-10 ***
## GroupID          0.97342
## SHOW_TRUST       0.02348 *
## GroupID:SHOW_TRUST 0.52757
## SHOW_ID          0.05556 .
## GroupID:SHOW_ID    0.42237
## SHOW_TRUST:SHOW_ID 0.01807 *
## GroupID:SHOW_TRUST:SHOW_ID 0.75861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "Analyze RECEIVER"
## [1] "Round number: 2"
## [1] "-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##          SS num Df Error SS den Df      F
## (Intercept)      15.6922      1  1.82293    21 180.7727
## GroupID          0.0491      4  1.82293    21  0.1415

```

```

## SHOW_TRUST          0.0456      1  0.94308      21  3.2313
## GroupID:SHOW_TRUST  0.0564      4  0.94308      21  0.3142
## SHOW_ID            0.4462      1  0.54586      21 13.8672
## GroupID:SHOW_ID     0.1287      4  0.54586      21  1.2379
## SHOW_TRUST:SHOW_ID  0.7592      1  1.06676      21 11.9861
## GroupID:SHOW_TRUST:SHOW_ID 0.2534      4  1.06676      21  1.2470
##                      Pr(>F)
## (Intercept)        8.702e-12 ***
## GroupID            0.96479
## SHOW_TRUST         0.14665
## GroupID:SHOW_TRUST 0.86525
## SHOW_ID            0.02041 *
## GroupID:SHOW_ID     0.32523
## SHOW_TRUST:SHOW_ID 0.02577 *
## GroupID:SHOW_TRUST:SHOW_ID 0.32177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "Analyze RECEIVER"
## [1] "Round number: 3"
## [1] "-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##                      SS num Df Error SS den Df      F
## (Intercept)        10.7198      1  0.66965      11 176.0886
## GroupID            0.0822      3  0.66965      11  0.4502
## SHOW_TRUST         0.3284      1  0.23443      11 11.5022
## GroupID:SHOW_TRUST 0.0857      3  0.23443      11  1.3397
## SHOW_ID            0.1271      1  0.38617      11 15.1525
## GroupID:SHOW_ID     0.0252      3  0.38617      11  0.2389
## SHOW_TRUST:SHOW_ID 0.3367      1  0.38919      11 10.2395
## GroupID:SHOW_TRUST:SHOW_ID 0.0986      3  0.38919      11  0.9293
##                      Pr(>F)
## (Intercept)        4.11e-08 ***
## GroupID            0.72225
## SHOW_TRUST         0.04273 *
## GroupID:SHOW_TRUST 0.31159
## SHOW_ID            0.03007 *
## GroupID:SHOW_ID     0.86744
## SHOW_TRUST:SHOW_ID 0.04934 *
## GroupID:SHOW_TRUST:SHOW_ID 0.45897
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----"
## [1] "Analyze RECEIVER"
## [1] "Round number: 4"
## [1] "-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##                      SS num Df Error SS den Df      F
## (Intercept)        15.6759      1  2.38555      19 124.8524
## GroupID            0.1339      4  2.38555      19  0.2665
## SHOW_TRUST         0.2743      1  0.33782      19 85.6685
## GroupID:SHOW_TRUST 0.0128      4  0.33782      19  0.1801
## SHOW_ID            0.1545      1  0.88764      19  7.9625
## GroupID:SHOW_ID     0.0776      4  0.88764      19  0.4154

```



```

## SHOW_TRUST:SHOW_ID          0.1778      1  0.50976      19  8.6846
## GroupID:SHOW_TRUST:SHOW_ID  0.0819      4  0.50976      19  0.7632
##                               Pr(>F)
## (Intercept)                 8.56e-10 ***
## GroupID                    0.8958083
## SHOW_TRUST                  0.0007576 ***
## GroupID:SHOW_TRUST          0.9458922
## SHOW_ID                    0.0477414 *
## GroupID:SHOW_ID             0.7954137
## SHOW_TRUST:SHOW_ID          0.0420959 *
## GroupID:SHOW_TRUST:SHOW_ID  0.5621206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "Analyze RECEIVER"
## [1] "Round number: 5"
## [1] "-----*-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##                               SS num Df Error SS den Df      F
## (Intercept)                 13.3054      1  1.03342      16 206.0029
## GroupID                    0.2484      4  1.03342      16  0.9613
## SHOW_TRUST                  0.0567      1  0.95196      16  3.6191
## GroupID:SHOW_TRUST          0.0627      4  0.95196      16  0.2633
## SHOW_ID                    0.1427      1  1.08276      16  3.5968
## GroupID:SHOW_ID             0.1587      4  1.08276      16  0.5863
## SHOW_TRUST:SHOW_ID          0.0155      1  0.80206      16  0.2174
## GroupID:SHOW_TRUST:SHOW_ID  0.2859      4  0.80206      16  1.4260
##                               Pr(>F)
## (Intercept)                 1.478e-10 ***
## GroupID                    0.4553
## SHOW_TRUST                  0.1299
## GroupID:SHOW_TRUST          0.8972
## SHOW_ID                    0.1308
## GroupID:SHOW_ID             0.6772
## SHOW_TRUST:SHOW_ID          0.6653
## GroupID:SHOW_TRUST:SHOW_ID  0.2706
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"

```

## Group effect analysis in each game

Date: Tue 6-Jun-2017

First we do send\_proportion ~ subject id, then run residual of the first regression ~ round\_number

```
## [1] "Analyze SENDER"
## [1] "Analyze game, Simple GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46272 -0.24091 -0.04615  0.18430  0.83133
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.12918    0.03602   3.586  0.00038 ***
## df$round_number -0.04306    0.01086  -3.965  8.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2974 on 373 degrees of freedom
## Multiple R-squared:  0.04044,    Adjusted R-squared:  0.03787
## F-statistic: 15.72 on 1 and 373 DF,  p-value: 8.805e-05
##
## [1] "Analyze game, ID GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6706 -0.2561 -0.0061  0.3071  0.6065
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.022628    0.038921  -0.581    0.561
## df$round_number  0.007543    0.011735   0.643    0.521
##
## Residual standard error: 0.3214 on 373 degrees of freedom
## Multiple R-squared:  0.001106,    Adjusted R-squared:  -0.001572
## F-statistic: 0.4131 on 1 and 373 DF,  p-value: 0.5208
##
## [1] "Analyze game, Score GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57804 -0.35558 -0.01663  0.43305  0.52034
##
## Coefficients:
```

```

##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.04606    0.04517  -1.020   0.309
## df$round_number 0.01535    0.01362   1.127   0.260
##
## Residual standard error: 0.373 on 373 degrees of freedom
## Multiple R-squared:  0.003395, Adjusted R-squared:  0.0007232
## F-statistic: 1.271 on 1 and 373 DF, p-value: 0.2604
##
## [1] "Analyze game, Combine GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60766 -0.33112 -0.02903  0.41046  0.51530
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.024253    0.044652  -0.543   0.587
## df$round_number 0.008084    0.013463   0.600   0.549
##
## Residual standard error: 0.3687 on 373 degrees of freedom
## Multiple R-squared:  0.0009657, Adjusted R-squared:  -0.001713
## F-statistic: 0.3606 on 1 and 373 DF, p-value: 0.5486
##
## [1] "Analyze RECEIVER"
## [1] "Analyze game, Simple GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28653 -0.25376 -0.04821  0.19500  0.74580
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.004784    0.037261  -0.128   0.898
## df$round_number 0.001689    0.011760   0.144   0.886
##
## Residual standard error: 0.2642 on 248 degrees of freedom
## Multiple R-squared:  8.322e-05, Adjusted R-squared:  -0.003949
## F-statistic: 0.02064 on 1 and 248 DF, p-value: 0.8859
##
## [1] "Analyze game, ID GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5142 -0.1211  0.0015  0.1679  0.5889
##

```

```

## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.047840   0.027736   1.725   0.0855 .
## df$round_number -0.016266   0.008491  -1.916   0.0563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2225 on 338 degrees of freedom
## Multiple R-squared:  0.01074,    Adjusted R-squared:  0.007814
## F-statistic:  3.67 on 1 and 338 DF,  p-value: 0.05625
##
## [1] "Analyze game, Score GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50159 -0.12783  0.01245  0.17541  0.54978
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0009071  0.0285047   0.032   0.975
## df$round_number -0.0003058  0.0086444  -0.035   0.972
##
## Residual standard error: 0.2242 on 322 degrees of freedom
## Multiple R-squared:  3.887e-06,    Adjusted R-squared:  -0.003102
## F-statistic: 0.001252 on 1 and 322 DF,  p-value: 0.9718
##
## [1] "Analyze game, Combine GAME"
##
## Call:
## lm(formula = l1.res ~ df$round_number)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51884 -0.11673  0.01007  0.14838  0.54884
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0016123  0.0254422   0.063   0.950
## df$round_number -0.0005468  0.0077672  -0.070   0.944
##
## Residual standard error: 0.2011 on 327 degrees of freedom
## Multiple R-squared:  1.516e-05,    Adjusted R-squared:  -0.003043
## F-statistic: 0.004957 on 1 and 327 DF,  p-value: 0.9439

```

## Regression of sending behavior on subjectID and round number

We test the regression power of predicting future sending proportion

```
## [1] "SIMPLE GAME with SENDER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44819 -0.10150 -0.02409  0.08454  0.94420
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.095631   0.073194   1.307 0.192495
## as.factor(subject_ids)2    0.029950   0.090957   0.329 0.742203
## as.factor(subject_ids)3    0.051086   0.085188   0.600 0.549221
## as.factor(subject_ids)4    0.011751   0.087447   0.134 0.893201
## as.factor(subject_ids)5    0.301155   0.100205   3.005 0.002905 **
## as.factor(subject_ids)6    0.115640   0.090773   1.274 0.203788
## as.factor(subject_ids)7    0.195979   0.090284   2.171 0.030836 *
## as.factor(subject_ids)8    0.109343   0.087777   1.246 0.213968
## as.factor(subject_ids)9    0.155162   0.090275   1.719 0.086817 .
## as.factor(subject_ids)10   0.543333   0.099774   5.446 1.17e-07 ***
## as.factor(subject_ids)11   0.116320   0.085343   1.363 0.174041
## as.factor(subject_ids)12  -0.001585   0.084294  -0.019 0.985008
## as.factor(subject_ids)13   0.164364   0.109645   1.499 0.135039
## as.factor(subject_ids)14   0.481565   0.092244   5.221 3.59e-07 ***
## as.factor(subject_ids)15   0.139155   0.083246   1.672 0.095771 .
## as.factor(subject_ids)16   0.495706   0.091631   5.410 1.40e-07 ***
## as.factor(subject_ids)17   0.301464   0.095316   3.163 0.001743 **
## as.factor(subject_ids)18   0.183104   0.092484   1.980 0.048747 *
## as.factor(subject_ids)19   0.208272   0.083550   2.493 0.013281 *
## as.factor(subject_ids)20   0.570032   0.100735   5.659 3.93e-08 ***
## as.factor(subject_ids)21   0.229096   0.088370   2.592 0.010054 *
## as.factor(subject_ids)22   0.174490   0.087404   1.996 0.046910 *
## as.factor(subject_ids)23   0.566533   0.101357   5.589 5.62e-08 ***
## as.factor(subject_ids)24   0.482778   0.096620   4.997 1.06e-06 ***
## as.factor(subject_ids)25   0.258038   0.089371   2.887 0.004204 **
## as.factor(subject_ids)26   0.075442   0.087243   0.865 0.387960
## as.factor(subject_ids)27   0.429862   0.093196   4.612 6.18e-06 ***
## as.factor(subject_ids)28   0.199092   0.092538   2.151 0.032337 *
## as.factor(subject_ids)29   0.322164   0.089833   3.586 0.000399 ***
## as.factor(subject_ids)30   0.767234   0.099953   7.676 3.10e-13 ***
## my_trusts          0.076441   0.081664   0.936 0.350095
## partner_trusts     0.069082   0.058462   1.182 0.238396
## round_numbers     -0.033655   0.010636  -3.164 0.001735 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2017 on 267 degrees of freedom
## Multiple R-squared:  0.56, Adjusted R-squared:  0.5072
```

```

## F-statistic: 10.62 on 32 and 267 DF,  p-value: < 2.2e-16
##
## [1] "SIMPLE GAME with RECEIVER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42127 -0.30664  0.03052  0.34521  1.54038
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.02193     0.21184  -4.824 2.37e-06 ***
## as.factor(subject_ids)2     0.30592     0.24029   1.273 0.204090
## as.factor(subject_ids)3     0.39634     0.26108   1.518 0.130176
## as.factor(subject_ids)4     0.06529     0.24728   0.264 0.791961
## as.factor(subject_ids)5    -0.12598     0.25820  -0.488 0.626013
## as.factor(subject_ids)6     0.48897     0.24164   2.024 0.044009 *
## as.factor(subject_ids)7     0.72483     0.25965   2.792 0.005624 **
## as.factor(subject_ids)8     0.41803     0.24802   1.685 0.093071 .
## as.factor(subject_ids)9     0.50569     0.24164   2.093 0.037318 *
## as.factor(subject_ids)10    0.47306     0.25579   1.849 0.065503 .
## as.factor(subject_ids)11    0.47120     0.25124   1.876 0.061811 .
## as.factor(subject_ids)12    0.76768     0.25079   3.061 0.002431 **
## as.factor(subject_ids)13    0.29483     0.22785   1.294 0.196806
## as.factor(subject_ids)14    0.24482     0.27040   0.905 0.366086
## as.factor(subject_ids)15    0.40293     0.26951   1.495 0.136079
## as.factor(subject_ids)16    0.28442     0.29739   0.956 0.339736
## as.factor(subject_ids)17    0.33519     0.25028   1.339 0.181635
## as.factor(subject_ids)18    0.30855     0.23806   1.296 0.196063
## as.factor(subject_ids)19    0.76776     0.27355   2.807 0.005374 **
## as.factor(subject_ids)20    0.61759     0.24463   2.525 0.012164 *
## as.factor(subject_ids)21    0.92524     0.25790   3.588 0.000397 ***
## as.factor(subject_ids)22    0.51435     0.26473   1.943 0.053073 .
## as.factor(subject_ids)23    0.78137     0.25362   3.081 0.002280 **
## as.factor(subject_ids)24    0.69753     0.24908   2.800 0.005476 **
## as.factor(subject_ids)25    0.61225     0.26436   2.316 0.021317 *
## as.factor(subject_ids)26    0.67841     0.24577   2.760 0.006173 **
## as.factor(subject_ids)27    0.63911     0.25136   2.543 0.011567 *
## as.factor(subject_ids)28    0.40888     0.23788   1.719 0.086805 .
## as.factor(subject_ids)29    0.67539     0.25513   2.647 0.008598 **
## as.factor(subject_ids)30    0.70961     0.28678   2.474 0.013969 *
## my_trusts         0.66908     0.19610   3.412 0.000745 ***
## partner_trusts     1.04953     0.14843   7.071 1.35e-11 ***
## round_numbers    -0.04083     0.02711  -1.506 0.133131
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.516 on 267 degrees of freedom
## Multiple R-squared:  0.4153, Adjusted R-squared:  0.3452
## F-statistic: 5.925 on 32 and 267 DF,  p-value: < 2.2e-16

```

```

## [1] "ID GAME with  SENDER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8537 -0.1226 -0.0003  0.1386  0.7152
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.18866    0.08800   2.144 0.032946 *
## as.factor(subject_ids)2 -0.25020    0.12152  -2.059 0.040482 *
## as.factor(subject_ids)3  0.06575    0.10754   0.611 0.541427
## as.factor(subject_ids)4  0.12906    0.10253   1.259 0.209195
## as.factor(subject_ids)5  0.24535    0.11641   2.108 0.035997 *
## as.factor(subject_ids)6  0.02802    0.11186   0.250 0.802420
## as.factor(subject_ids)7  0.05410    0.13440   0.403 0.687621
## as.factor(subject_ids)8  0.05745    0.10622   0.541 0.589032
## as.factor(subject_ids)9 -0.17318    0.10287  -1.683 0.093455 .
## as.factor(subject_ids)10  0.02059    0.09917   0.208 0.835655
## as.factor(subject_ids)11  0.12844    0.11294   1.137 0.256481
## as.factor(subject_ids)12 -0.04771    0.11481  -0.416 0.678101
## as.factor(subject_ids)13  0.03726    0.10120   0.368 0.713028
## as.factor(subject_ids)14  0.04870    0.11527   0.422 0.673034
## as.factor(subject_ids)15  0.22365    0.10734   2.084 0.038156 *
## as.factor(subject_ids)16  0.04604    0.10345   0.445 0.656610
## as.factor(subject_ids)17  0.41700    0.11862   3.515 0.000516 ***
## as.factor(subject_ids)18  0.45466    0.11233   4.048 6.78e-05 ***
## as.factor(subject_ids)19  0.07740    0.10490   0.738 0.461219
## as.factor(subject_ids)20  0.20043    0.10879   1.842 0.066520 .
## as.factor(subject_ids)21  0.19458    0.11080   1.756 0.080230 .
## as.factor(subject_ids)22  0.06247    0.11478   0.544 0.586747
## as.factor(subject_ids)23  0.19416    0.10531   1.844 0.066344 .
## as.factor(subject_ids)24  0.19273    0.10506   1.834 0.067697 .
## as.factor(subject_ids)25 -0.07426    0.10336  -0.718 0.473130
## as.factor(subject_ids)26 -0.06606    0.11277  -0.586 0.558499
## as.factor(subject_ids)27 -0.11862    0.10289  -1.153 0.249993
## as.factor(subject_ids)28 -0.09967    0.10877  -0.916 0.360319
## as.factor(subject_ids)29  0.06943    0.11485   0.605 0.546024
## as.factor(subject_ids)30  0.18283    0.11410   1.602 0.110271
## my_trusts          0.49327    0.10041   4.913 1.57e-06 ***
## partner_trusts      0.49049    0.07745   6.333 1.01e-09 ***
## round_numbers      -0.04462    0.01455  -3.067 0.002384 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2551 on 267 degrees of freedom
## Multiple R-squared:  0.5034, Adjusted R-squared:  0.4439
## F-statistic: 8.458 on 32 and 267 DF,  p-value: < 2.2e-16
##
## [1] "ID GAME with  RECEIVER"
##

```

```
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49373 -0.11072  0.04535  0.20213  1.11041
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.03875    0.19130   0.203  0.83963
## as.factor(subject_ids)2 -0.27187    0.21530  -1.263  0.20778
## as.factor(subject_ids)3 -0.06407    0.21335  -0.300  0.76419
## as.factor(subject_ids)4 -0.09552    0.22147  -0.431  0.66659
## as.factor(subject_ids)5 -0.27676    0.21128  -1.310  0.19135
## as.factor(subject_ids)6 -0.12213    0.20986  -0.582  0.56110
## as.factor(subject_ids)7  0.33493    0.19723   1.698  0.09065 .
## as.factor(subject_ids)8  0.28752    0.21646   1.328  0.18522
## as.factor(subject_ids)9  0.19444    0.23311   0.834  0.40496
## as.factor(subject_ids)10 0.44571    0.23975   1.859  0.06412 .
## as.factor(subject_ids)11 0.22837    0.21000   1.087  0.27782
## as.factor(subject_ids)12 0.08997    0.20435   0.440  0.66008
## as.factor(subject_ids)13 0.19268    0.22887   0.842  0.40060
## as.factor(subject_ids)14 0.16560    0.20493   0.808  0.41975
## as.factor(subject_ids)15 -0.13683    0.21115  -0.648  0.51754
## as.factor(subject_ids)16 0.32190    0.22391   1.438  0.15171
## as.factor(subject_ids)17 0.09681    0.21388   0.453  0.65119
## as.factor(subject_ids)18 0.39201    0.20771   1.887  0.06020 .
## as.factor(subject_ids)19 0.24288    0.21636   1.123  0.26263
## as.factor(subject_ids)20 0.19568    0.21489   0.911  0.36332
## as.factor(subject_ids)21 0.23341    0.20823   1.121  0.26333
## as.factor(subject_ids)22 0.13149    0.20445   0.643  0.52069
## as.factor(subject_ids)23 0.20384    0.21804   0.935  0.35068
## as.factor(subject_ids)24 0.32206    0.21618   1.490  0.13746
## as.factor(subject_ids)25 -0.24092    0.22134  -1.088  0.27738
## as.factor(subject_ids)26 -0.01231    0.21216  -0.058  0.95377
## as.factor(subject_ids)27 0.05652    0.22319   0.253  0.80028
## as.factor(subject_ids)28 -0.07579    0.21571  -0.351  0.72559
## as.factor(subject_ids)29 0.15100    0.20413   0.740  0.46012
## as.factor(subject_ids)30 0.18590    0.20806   0.893  0.37241
## my_trusts          0.52388    0.18343   2.856  0.00463 **
## partner_trusts     0.68946    0.12555   5.492 9.27e-08 ***
## round_numbers     -0.11197    0.02469  -4.534 8.73e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4269 on 267 degrees of freedom
## Multiple R-squared:  0.348, Adjusted R-squared:  0.2698
## F-statistic: 4.453 on 32 and 267 DF, p-value: 3.719e-12
## [1] "SCORE GAME with SENDER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
```



```

##      round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6213 -0.1095  0.0022  0.1224  0.9764
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.01266    0.07931  -0.160  0.873273
## as.factor(subject_ids)2 -0.23262    0.09337  -2.491  0.013334 *
## as.factor(subject_ids)3 -0.05136    0.09200  -0.558  0.577138
## as.factor(subject_ids)4 -0.23275    0.09597  -2.425  0.015958 *
## as.factor(subject_ids)5  0.21251    0.08718   2.438  0.015435 *
## as.factor(subject_ids)6 -0.22646    0.09018  -2.511  0.012627 *
## as.factor(subject_ids)7 -0.20870    0.08567  -2.436  0.015502 *
## as.factor(subject_ids)8 -0.04438    0.09573  -0.464  0.643343
## as.factor(subject_ids)9 -0.34158    0.09268  -3.686  0.000276 ***
## as.factor(subject_ids)10 -0.06511    0.08594  -0.758  0.449290
## as.factor(subject_ids)11  0.17227    0.09636   1.788  0.074951 .
## as.factor(subject_ids)12 -0.07208    0.09208  -0.783  0.434464
## as.factor(subject_ids)13  0.19998    0.08653   2.311  0.021581 *
## as.factor(subject_ids)14  0.16434    0.09338   1.760  0.079578 .
## as.factor(subject_ids)15  0.06170    0.08604   0.717  0.473943
## as.factor(subject_ids)16  0.14031    0.08946   1.568  0.117986
## as.factor(subject_ids)17  0.17492    0.09638   1.815  0.070660 .
## as.factor(subject_ids)18  0.19105    0.09206   2.075  0.038916 *
## as.factor(subject_ids)19  0.06895    0.11056   0.624  0.533363
## as.factor(subject_ids)20 -0.14028    0.09177  -1.529  0.127536
## as.factor(subject_ids)21  0.13560    0.08371   1.620  0.106432
## as.factor(subject_ids)22  0.13373    0.08243   1.622  0.105914
## as.factor(subject_ids)23  0.06903    0.09232   0.748  0.455281
## as.factor(subject_ids)24  0.06794    0.09515   0.714  0.475832
## as.factor(subject_ids)25  0.11207    0.11073   1.012  0.312431
## as.factor(subject_ids)26 -0.19448    0.09095  -2.138  0.033402 *
## as.factor(subject_ids)27 -0.15020    0.08457  -1.776  0.076871 .
## as.factor(subject_ids)28 -0.24872    0.08603  -2.891  0.004156 **
## as.factor(subject_ids)29 -0.07043    0.09230  -0.763  0.446102
## as.factor(subject_ids)30  0.01818    0.09586   0.190  0.849743
## my_trusts      0.07962    0.07354   1.083  0.279878
## partner_trusts 1.02318    0.06216  16.461 < 2e-16 ***
## round_numbers  0.01595    0.01109   1.438  0.151662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2043 on 267 degrees of freedom
## Multiple R-squared:  0.7458, Adjusted R-squared:  0.7153
## F-statistic: 24.48 on 32 and 267 DF,  p-value: < 2.2e-16
##
## [1] "SCORE GAME with RECEIVER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##      round_numbers)
##

```

```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.52743 -0.15453  0.05119  0.24646  1.19539
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.017278   0.188336  -0.092  0.926974
## as.factor(subject_ids)2 -0.789159   0.216418  -3.646  0.000320 ***
## as.factor(subject_ids)3 -0.213550   0.206360  -1.035  0.301680
## as.factor(subject_ids)4 -0.247437   0.202797  -1.220  0.223496
## as.factor(subject_ids)5 -0.234469   0.216391  -1.084  0.279545
## as.factor(subject_ids)6 -0.671299   0.222630  -3.015  0.002814 **
## as.factor(subject_ids)7 -0.227873   0.235318  -0.968  0.333739
## as.factor(subject_ids)8  0.130399   0.202885   0.643  0.520953
## as.factor(subject_ids)9 -0.055036   0.219833  -0.250  0.802507
## as.factor(subject_ids)10 0.079731   0.223045   0.357  0.721026
## as.factor(subject_ids)11 -0.003325   0.202638  -0.016  0.986919
## as.factor(subject_ids)12 0.176456   0.209631   0.842  0.400682
## as.factor(subject_ids)13 -0.276699   0.223407  -1.239  0.216603
## as.factor(subject_ids)14 -0.117150   0.206224  -0.568  0.570462
## as.factor(subject_ids)15 -0.198460   0.223943  -0.886  0.376304
## as.factor(subject_ids)16 0.121538   0.211571   0.574  0.566142
## as.factor(subject_ids)17 -0.128686   0.204872  -0.628  0.530454
## as.factor(subject_ids)18 -0.117172   0.206680  -0.567  0.571241
## as.factor(subject_ids)19 -0.044654   0.194342  -0.230  0.818447
## as.factor(subject_ids)20 -0.276578   0.224015  -1.235  0.218049
## as.factor(subject_ids)21 -0.017967   0.231657  -0.078  0.938239
## as.factor(subject_ids)22 0.078809   0.241644   0.326  0.744576
## as.factor(subject_ids)23 -0.145894   0.206890  -0.705  0.481316
## as.factor(subject_ids)24 0.001947   0.202800   0.010  0.992349
## as.factor(subject_ids)25 0.099958   0.193304   0.517  0.605513
## as.factor(subject_ids)26 -0.883194   0.226761  -3.895  0.000124 ***
## as.factor(subject_ids)27 -0.442123   0.232513  -1.901  0.058314 .
## as.factor(subject_ids)28 -0.730742   0.250204  -2.921  0.003792 **
## as.factor(subject_ids)29 -0.294889   0.207373  -1.422  0.156186
## as.factor(subject_ids)30 -0.279031   0.202379  -1.379  0.169124
## my_trusts       0.760906   0.168931   4.504  9.97e-06 ***
## partner_trusts  0.290496   0.142151   2.044  0.041976 *
## round_numbers   -0.010289   0.024780  -0.415  0.678322
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.458 on 267 degrees of freedom
## Multiple R-squared:  0.4405, Adjusted R-squared:  0.3735
## F-statistic:  6.57 on 32 and 267 DF,  p-value: < 2.2e-16

## [1] "COMBINE GAME with SENDER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max

```

```

## -0.67728 -0.11400 0.00588 0.10843 0.92274
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.069774   0.106775   0.653  0.5140
## as.factor(subject_ids)2 -0.304711   0.117775  -2.587  0.0102 *
## as.factor(subject_ids)3 -0.164314   0.113828  -1.444  0.1500
## as.factor(subject_ids)4 -0.002060   0.112682  -0.018  0.9854
## as.factor(subject_ids)5  0.024473   0.121249   0.202  0.8402
## as.factor(subject_ids)6 -0.111419   0.124535  -0.895  0.3718
## as.factor(subject_ids)7 -0.207214   0.116109  -1.785  0.0755 .
## as.factor(subject_ids)8  0.118001   0.121868   0.968  0.3338
## as.factor(subject_ids)9 -0.238743   0.118654  -2.012  0.0452 *
## as.factor(subject_ids)10 0.012295   0.118748   0.104  0.9176
## as.factor(subject_ids)11 0.073878   0.125825   0.587  0.5576
## as.factor(subject_ids)12 -0.159330   0.120599  -1.321  0.1876
## as.factor(subject_ids)13 0.230069   0.119475   1.926  0.0552 .
## as.factor(subject_ids)14 0.152321   0.120662   1.262  0.2079
## as.factor(subject_ids)15 0.199192   0.122421   1.627  0.1049
## as.factor(subject_ids)16 -0.005169   0.126614  -0.041  0.9675
## as.factor(subject_ids)17 0.252633   0.119999   2.105  0.0362 *
## as.factor(subject_ids)18 0.156076   0.120444   1.296  0.1961
## as.factor(subject_ids)19 -0.035566   0.115674  -0.307  0.7587
## as.factor(subject_ids)20 -0.245708   0.121213  -2.027  0.0436 *
## as.factor(subject_ids)21 0.200783   0.116790   1.719  0.0867 .
## as.factor(subject_ids)22 0.123132   0.119317   1.032  0.3030
## as.factor(subject_ids)23 0.046890   0.123690   0.379  0.7049
## as.factor(subject_ids)24 0.052480   0.121402   0.432  0.6659
## as.factor(subject_ids)25 0.111539   0.113852   0.980  0.3281
## as.factor(subject_ids)26 -0.136684   0.124332  -1.099  0.2726
## as.factor(subject_ids)27 -0.173349   0.118476  -1.463  0.1446
## as.factor(subject_ids)28 -0.126613   0.115717  -1.094  0.2749
## as.factor(subject_ids)29 -0.122831   0.123601  -0.994  0.3212
## as.factor(subject_ids)30 0.212356   0.122222   1.737  0.0835 .
## my_trusts          0.413093   0.087380   4.728 3.68e-06 ***
## partner_trusts      0.733572   0.066999  10.949 < 2e-16 ***
## round_numbers      -0.012912   0.011734  -1.100  0.2721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2159 on 267 degrees of freedom
## Multiple R-squared:  0.7155, Adjusted R-squared:  0.6814
## F-statistic: 20.98 on 32 and 267 DF, p-value: < 2.2e-16
##
## [1] "COMBINE GAME with RECEIVER"
##
## Call:
## lm(formula = sends ~ as.factor(subject_ids) + my_trusts + partner_trusts +
##     round_numbers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43873 -0.12033  0.02294  0.20087  1.70184
##

```

```

## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.197554   0.151220  -1.306 0.192540
## as.factor(subject_ids)2 -0.690598   0.185597  -3.721 0.000242 ***
## as.factor(subject_ids)3 -0.156138   0.192564  -0.811 0.418181
## as.factor(subject_ids)4 -0.263992   0.199086  -1.326 0.185966
## as.factor(subject_ids)5  0.124720   0.162386   0.768 0.443136
## as.factor(subject_ids)6 -0.249088   0.177627  -1.402 0.161985
## as.factor(subject_ids)7  0.324625   0.186983   1.736 0.083697 .
## as.factor(subject_ids)8  0.349449   0.161985   2.157 0.031875 *
## as.factor(subject_ids)9 -0.178636   0.198474  -0.900 0.368906
## as.factor(subject_ids)10 0.369866   0.165913   2.229 0.026628 *
## as.factor(subject_ids)11 0.168146   0.158013   1.064 0.288231
## as.factor(subject_ids)12 0.233112   0.166212   1.402 0.161929
## as.factor(subject_ids)13 -0.069137   0.172324  -0.401 0.688590
## as.factor(subject_ids)14 0.049845   0.167079   0.298 0.765681
## as.factor(subject_ids)15 -0.154056   0.163724  -0.941 0.347582
## as.factor(subject_ids)16 0.157366   0.159050   0.989 0.323358
## as.factor(subject_ids)17 0.195773   0.172322   1.136 0.256940
## as.factor(subject_ids)18 0.008483   0.172665   0.049 0.960852
## as.factor(subject_ids)19 0.225238   0.177848   1.266 0.206451
## as.factor(subject_ids)20 0.017480   0.165985   0.105 0.916210
## as.factor(subject_ids)21 -0.160388   0.177828  -0.902 0.367908
## as.factor(subject_ids)22 -0.115233   0.170409  -0.676 0.499489
## as.factor(subject_ids)23 -0.146699   0.158935  -0.923 0.356835
## as.factor(subject_ids)24 0.164351   0.162229   1.013 0.311938
## as.factor(subject_ids)25 0.441125   0.187955   2.347 0.019658 *
## as.factor(subject_ids)26 0.243746   0.162703   1.498 0.135287
## as.factor(subject_ids)27 0.306916   0.168848   1.818 0.070229 .
## as.factor(subject_ids)28 0.121514   0.183329   0.663 0.508019
## as.factor(subject_ids)29 0.199026   0.165466   1.203 0.230109
## as.factor(subject_ids)30 0.233026   0.161592   1.442 0.150457
## my_trusts       0.797014   0.171133   4.657 5.06e-06 ***
## partner_trusts  0.666376   0.133857   4.978 1.15e-06 ***
## round_numbers   -0.073437   0.022223  -3.305 0.001081 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4056 on 267 degrees of freedom
## Multiple R-squared:  0.5048, Adjusted R-squared:  0.4455
## F-statistic: 8.506 on 32 and 267 DF,  p-value: < 2.2e-16

```

## Reputation score

We applied linear regression on reputation score to see if the reputation score should be used instead of trust score.

Reputation score = average of previous send proportion

```
## [1] "Linear regression of relative sending on reputation value of Simple Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##     partner_reputation, data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.157174 -0.049438 -0.000103  0.048766  0.142786
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.24352    0.06529  -3.730  0.000988 ***
## my_reputation    0.26260    0.23220   1.131  0.268815
## my_trust_value   1.13615    0.35142   3.233  0.003426 **
## trust_value      0.01123    0.27764   0.040  0.968056
## partner_reputation 0.24217    0.21208   1.142  0.264332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08406 on 25 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8576
## F-statistic: 44.67 on 4 and 25 DF,  p-value: 4.906e-11
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17234 -0.04094 -0.01668  0.04963  0.18601
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.3067    0.1438  -2.133  0.042496 *
## my_trust_value    1.6318    0.4227   3.860  0.000672 ***
## trust_value      0.4005    0.4393   0.912  0.370256
## my_trust_value:trust_value -0.4468    1.2688  -0.352  0.727575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08644 on 26 degrees of freedom
## Multiple R-squared:  0.865, Adjusted R-squared:  0.8494
## F-statistic: 55.54 on 3 and 26 DF,  p-value: 1.934e-11
##
## [1] "Test the interaction variable alone"
```

```
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19027 -0.05911 -0.01196  0.05886  0.14780
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.18520    0.04184  -4.426 0.000133 ***
## my_trust_value  1.50691    0.11930  12.631 4.4e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08759 on 28 degrees of freedom
## Multiple R-squared:  0.8507, Adjusted R-squared:  0.8454
## F-statistic: 159.6 on 1 and 28 DF,  p-value: 4.401e-13
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18535 -0.07639 -0.03368  0.03254  0.35604
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.01458    0.05195  -0.281  0.781
## combine_trust  3.03579    0.43499   6.979 1.37e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.137 on 28 degrees of freedom
## Multiple R-squared:  0.635, Adjusted R-squared:  0.6219
## F-statistic: 48.71 on 1 and 28 DF,  p-value: 1.373e-07
##
## [1] "Linear regression of relative sending on reputation value of ID Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##      partner_reputation, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17819 -0.08414  0.01242  0.05238  0.24874
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40430    0.22613  -1.788  0.0859 .
## my_reputation  0.13423    0.46406   0.289  0.7748
## my_trust_value  1.30637    0.59263   2.204  0.0369 *
## trust_value    0.53231    0.36297   1.467  0.1550
```

```

## partner_reputation 0.09122 0.36980 0.247 0.8072
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.105 on 25 degrees of freedom
## Multiple R-squared: 0.7653, Adjusted R-squared: 0.7278
## F-statistic: 20.38 on 4 and 25 DF, p-value: 1.429e-07
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
## my_trust_value, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17352 -0.07100  0.01394  0.05352  0.23447
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.0001726  0.5356202   0.000   1.000
## my_trust_value      0.6192925  1.1591418   0.534   0.598
## trust_value      -0.2322279  1.1241335  -0.207   0.838
## my_trust_value:trust_value  1.7740167  2.4476382   0.725   0.475
##
## Residual standard error: 0.1023 on 26 degrees of freedom
## Multiple R-squared: 0.7681, Adjusted R-squared: 0.7414
## F-statistic: 28.71 on 3 and 26 DF, p-value: 2.076e-08
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23128 -0.05684  0.00633  0.05522  0.22734
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.10361    0.07426  -1.395   0.174
## my_trust_value  1.42051    0.16027   8.863 1.29e-09 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.105 on 28 degrees of freedom
## Multiple R-squared: 0.7372, Adjusted R-squared: 0.7278
## F-statistic: 78.56 on 1 and 28 DF, p-value: 1.287e-09
##
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_id)
##
## Residuals:

```

```

##      Min      1Q   Median      3Q      Max
## -0.18545 -0.08686 -0.00256  0.04153  0.23345
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.03935    0.07116  -0.553   0.585
## combine_trust  2.81050    0.33578   8.370 4.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1094 on 28 degrees of freedom
## Multiple R-squared:  0.7145, Adjusted R-squared:  0.7043
## F-statistic: 70.06 on 1 and 28 DF,  p-value: 4.182e-09
##
## [1] "Linear regression of relative sending on reputation value of Score Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##     partner_reputation, data = x_score)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -0.176618 -0.049620  0.007802  0.062108  0.128669
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.6261    0.1019  -6.146 2.00e-06 ***
## my_reputation     0.1678    0.2636   0.636  0.53034
## my_trust_value     1.1877    0.3681   3.227  0.00348 **
## trust_value        1.4762    0.3178   4.645 9.34e-05 ***
## partner_reputation -0.2527    0.2733  -0.925  0.36391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08728 on 25 degrees of freedom
## Multiple R-squared:  0.893, Adjusted R-squared:  0.8758
## F-statistic: 52.14 on 4 and 25 DF,  p-value: 8.997e-12
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_score)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -0.20494 -0.06274  0.01232  0.07127  0.13125
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.1992    0.4480  -0.445   0.660
## my_trust_value     0.3970    0.9083   0.437   0.666
## trust_value        0.1731    1.0874   0.159   0.875
## my_trust_value:trust_value  2.2904    2.1235   1.079   0.291

```



```

##
## Residual standard error: 0.0857 on 26 degrees of freedom
## Multiple R-squared:  0.8927, Adjusted R-squared:  0.8803
## F-statistic: 72.09 on 3 and 26 DF,  p-value: 9.938e-13
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30783 -0.08080 -0.02321  0.07674  0.28592
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.3395     0.0981  -3.461  0.00175 **
## my_trust_value  1.9857     0.2146   9.255 5.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1251 on 28 degrees of freedom
## Multiple R-squared:  0.7536, Adjusted R-squared:  0.7448
## F-statistic: 85.65 on 1 and 28 DF,  p-value: 5.173e-10
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.187436 -0.057111  0.003304  0.067821  0.133944
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.08017     0.04415  -1.816  0.0802 .
## combine_trust  2.94130     0.19542  15.051 6e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08361 on 28 degrees of freedom
## Multiple R-squared:  0.89, Adjusted R-squared:  0.8861
## F-statistic: 226.5 on 1 and 28 DF,  p-value: 5.995e-15
##
## [1] "Linear regression of relative sending on reputation value of Combine Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##      partner_reputation, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16324 -0.04152 -0.00388  0.03458  0.17991
##

```

```

## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.59209    0.09153  -6.469 8.95e-07 ***
## my_reputation    0.40566    0.32551   1.246 0.22424
## my_trust_value   0.91386    0.50707   1.802 0.08358 .
## trust_value     1.25835    0.39763   3.165 0.00405 **
## partner_reputation -0.12418    0.27024  -0.460 0.64984
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08986 on 25 degrees of freedom
## Multiple R-squared:  0.9029, Adjusted R-squared:  0.8874
## F-statistic: 58.14 on 4 and 25 DF,  p-value: 2.677e-12
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.151630 -0.044313  0.002645  0.038603  0.198590
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.61887    0.36917  -1.676 0.1056
## my_trust_value    1.47851    0.80055   1.847 0.0762 .
## trust_value      1.05164    0.84718   1.241 0.2256
## my_trust_value:trust_value 0.03711    1.71567   0.022 0.9829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09093 on 26 degrees of freedom
## Multiple R-squared:  0.8966, Adjusted R-squared:  0.8847
## F-statistic: 75.19 on 3 and 26 DF,  p-value: 6.11e-13
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19143 -0.07747 -0.02447  0.04625  0.31950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.36971    0.08619  -4.289 0.000193 ***
## my_trust_value  2.02721    0.18413  11.010 1.1e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1181 on 28 degrees of freedom

```

```

## Multiple R-squared:  0.8124, Adjusted R-squared:  0.8057
## F-statistic: 121.2 on 1 and 28 DF,  p-value: 1.104e-11
##
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.210693 -0.048904 -0.001717  0.052600  0.227406
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.06094    0.04602  -1.324   0.196
## combine_trust  2.79971    0.19576  14.302 2.14e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09457 on 28 degrees of freedom
## Multiple R-squared:  0.8796, Adjusted R-squared:  0.8753
## F-statistic: 204.5 on 1 and 28 DF,  p-value: 2.135e-14
##
## [1] "Linear regression of relative sending on reputation value of Simple Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##     partner_reputation + AbsPartnerSend, data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23176 -0.10296  0.01343  0.08570  0.17720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.02336    0.13015   0.179   0.8591
## my_reputation  -0.75347    0.31847  -2.366   0.0264 *
## my_trust_value   2.11822    0.44393   4.772 7.42e-05 ***
## trust_value      0.15140    0.51837   0.292   0.7727
## partner_reputation -0.56442    0.23007  -2.453   0.0218 *
## AbsPartnerSend  -0.01885    0.02983  -0.632   0.5335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1246 on 24 degrees of freedom
## Multiple R-squared:  0.7287, Adjusted R-squared:  0.6722
## F-statistic: 12.89 on 5 and 24 DF,  p-value: 3.745e-06
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_simple)
##
## Residuals:

```

```

##      Min      1Q   Median      3Q      Max
## -0.21443 -0.09409 -0.03086  0.06791  0.48515
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.1450     0.3458   0.419   0.678
## my_trust_value        0.9858     0.8580   1.149   0.261
## trust_value          -0.5812     0.9106  -0.638   0.529
## my_trust_value:trust_value  0.1236     2.3039   0.054   0.958
##
## Residual standard error: 0.1566 on 26 degrees of freedom
## Multiple R-squared:  0.5361, Adjusted R-squared:  0.4825
## F-statistic: 10.01 on 3 and 26 DF,  p-value: 0.0001456
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_simple)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -0.26311 -0.09937 -0.02062  0.04804  0.41756
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.08705     0.07678  -1.134   0.267
## my_trust_value  1.05048     0.20281   5.180 1.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1583 on 28 degrees of freedom
## Multiple R-squared:  0.4893, Adjusted R-squared:  0.4711
## F-statistic: 26.83 on 1 and 28 DF,  p-value: 1.697e-05
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_simple)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -0.33242 -0.11733 -0.03566  0.08618  0.51256
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.05767     0.08832   0.653  0.51910
## combine_trust  1.64847     0.59481   2.771  0.00981 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1962 on 28 degrees of freedom
## Multiple R-squared:  0.2153, Adjusted R-squared:  0.1872
## F-statistic: 7.681 on 1 and 28 DF,  p-value: 0.009806
##
## [1] "Linear regression of relative sending on reputation value of ID Game for type: RECEIVER"

```

```
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + partner_reputation +
##      trust_value + partner_reputation + AbsPartnerSend, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.189929 -0.059582 -0.009515  0.057207  0.140608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.14989    0.17625   -0.850   0.4035
## my_reputation     0.29679    0.31230    0.950   0.3514
## my_trust_value     0.82827    0.46330    1.788   0.0864 .
## partner_reputation -0.46524    0.35805   -1.299   0.2062
## trust_value       0.94534    0.54589    1.732   0.0962 .
## AbsPartnerSend   -0.02509    0.01634   -1.535   0.1378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08328 on 24 degrees of freedom
## Multiple R-squared:  0.7942, Adjusted R-squared:  0.7514
## F-statistic: 18.53 on 5 and 24 DF,  p-value: 1.524e-07
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##      my_trust_value, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.193188 -0.060362 -0.006606  0.048345  0.160393
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.1266     0.8340   0.152   0.880
## my_trust_value     0.7566     1.7421   0.434   0.668
## trust_value      -0.5395     1.8217  -0.296   0.769
## my_trust_value:trust_value  0.9979     3.7867   0.264   0.794
##
## Residual standard error: 0.08945 on 26 degrees of freedom
## Multiple R-squared:  0.7428, Adjusted R-squared:  0.7132
## F-statistic: 25.03 on 3 and 26 DF,  p-value: 7.856e-08
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.202381 -0.061546 -0.005367  0.050570  0.160479
##
```

```

## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.11864    0.06455  -1.838   0.0767 .
## my_trust_value  1.21060    0.13498   8.968 1.01e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08637 on 28 degrees of freedom
## Multiple R-squared:  0.7418, Adjusted R-squared:  0.7326
## F-statistic: 80.43 on 1 and 28 DF,  p-value: 1.006e-09
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.275334 -0.077234  0.001101  0.056730  0.209438
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.02147    0.06900  -0.311   0.758
## combine_trust  2.16006    0.30900   6.991 1.33e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1026 on 28 degrees of freedom
## Multiple R-squared:  0.6357, Adjusted R-squared:  0.6227
## F-statistic: 48.87 on 1 and 28 DF,  p-value: 1.333e-07
##
## [1] "Linear regression of relative sending on reputation value of Score Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##     partner_reputation + AbsPartnerSend, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20660 -0.06624 -0.01861  0.07399  0.21424
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.11867    0.23255   0.510   0.6145
## my_reputation   0.70664    0.32781   2.156   0.0414 *
## my_trust_value   0.48724    0.52241   0.933   0.3603
## trust_value     0.42296    0.69390   0.610   0.5479
## partner_reputation -0.49308    0.42937  -1.148   0.2621
## AbsPartnerSend  -0.03424    0.02819  -1.215   0.2363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1176 on 24 degrees of freedom
## Multiple R-squared:  0.58, Adjusted R-squared:  0.4924
## F-statistic: 6.627 on 5 and 24 DF,  p-value: 0.0005246

```

```
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.260785 -0.052235  0.004605  0.057604  0.203733
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.2691     0.9330  -0.288   0.775
## my_trust_value     2.0338     1.8371   1.107   0.278
## trust_value       0.3841     2.0837   0.184   0.855
## my_trust_value:trust_value -1.9629     3.9906  -0.492   0.627
##
## Residual standard error: 0.1308 on 26 degrees of freedom
## Multiple R-squared:  0.4371, Adjusted R-squared:  0.3722
## F-statistic:  6.73 on 3 and 26 DF,  p-value: 0.001645
##
## [1] "Test the interaction variable alone"
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27229 -0.08560  0.02484  0.07180  0.19451
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.02284     0.11234  -0.203  0.840384
## my_trust_value  0.97933     0.22815   4.292 0.000191 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1305 on 28 degrees of freedom
## Multiple R-squared:  0.3969, Adjusted R-squared:  0.3753
## F-statistic: 18.43 on 1 and 28 DF,  p-value: 0.000191
##
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26457 -0.09359  0.03418  0.10083  0.21963
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.17591     0.09227   1.906  0.06690 .
## combine_trust  1.22611     0.39764   3.083  0.00456 **
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1452 on 28 degrees of freedom
## Multiple R-squared:  0.2535, Adjusted R-squared:  0.2268
## F-statistic: 9.508 on 1 and 28 DF,  p-value: 0.004565
##
## [1] "Linear regression of relative sending on reputation value of Combine Game for type: RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_reputation + my_trust_value + trust_value +
##     partner_reputation + AbsPartnerSend, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.142209 -0.055860  0.004107  0.054064  0.153756
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.48253    0.15847   3.045  0.00558 **
## my_reputation      0.60957    0.37615   1.621  0.11818
## my_trust_value    -0.07530    0.59039  -0.128  0.89957
## trust_value       -0.66498    0.37269  -1.784  0.08703 .
## partner_reputation -0.17985    0.30913  -0.582  0.56613
## AbsPartnerSend     0.01853    0.02263   0.819  0.42094
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09297 on 24 degrees of freedom
## Multiple R-squared:  0.5913, Adjusted R-squared:  0.5062
## F-statistic: 6.945 on 5 and 24 DF,  p-value: 0.0003873
##
## [1] "Test the interaction"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + trust_value *
##     my_trust_value, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17384 -0.05200 -0.01441  0.06781  0.14184
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.1550    0.3669   0.422   0.676
## my_trust_value      1.2526    0.7800   1.606   0.120
## trust_value       -0.2445    0.8036  -0.304   0.763
## my_trust_value:trust_value -0.7821    1.6399  -0.477   0.637
##
## Residual standard error: 0.09432 on 26 degrees of freedom
## Multiple R-squared:  0.5443, Adjusted R-squared:  0.4917
## F-statistic: 10.35 on 3 and 26 DF,  p-value: 0.0001161
##
## [1] "Test the interaction variable alone"

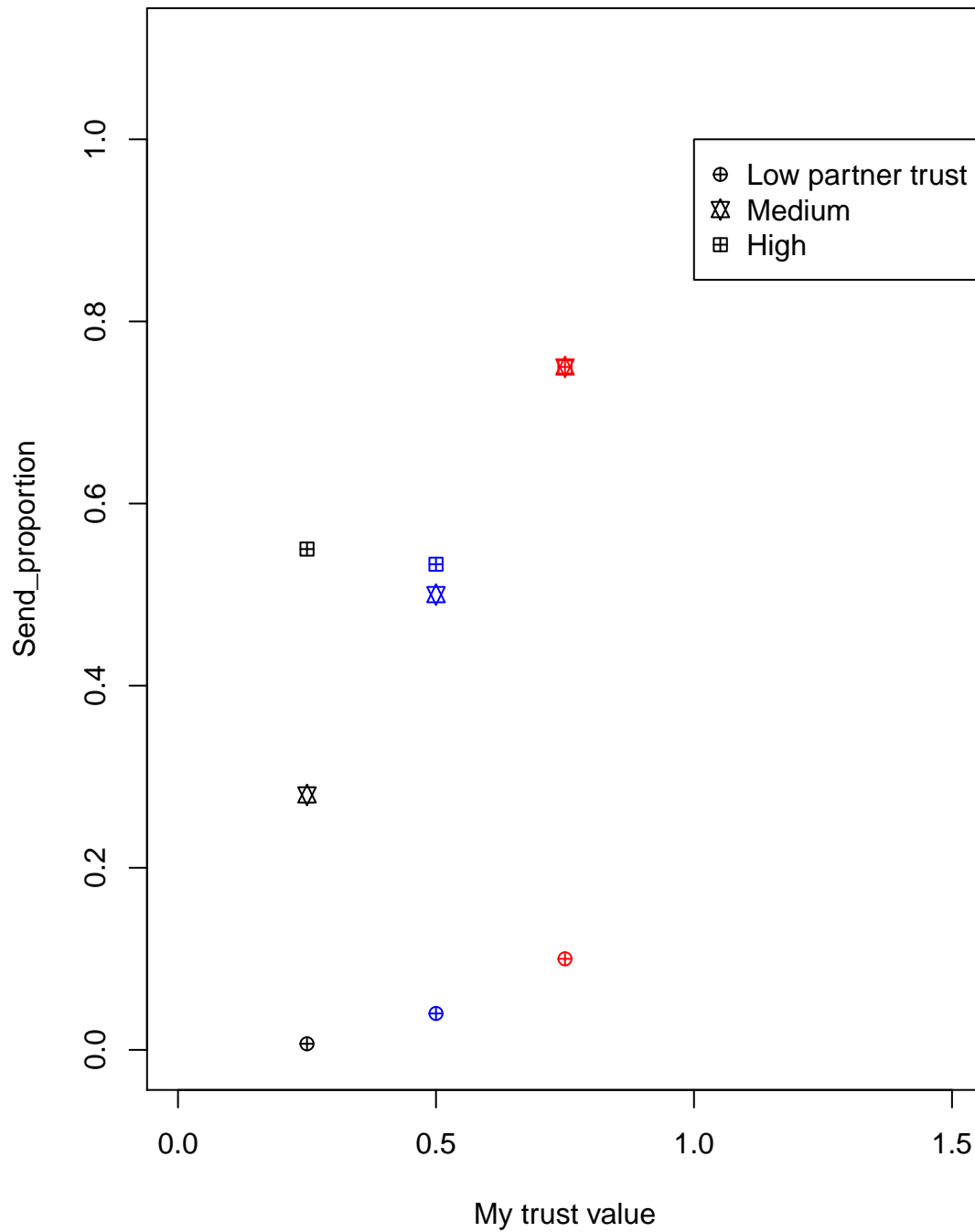
```



```
##
## Call:
## lm(formula = RelSend ~ my_trust_value, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24893 -0.05886  0.04055  0.06545  0.15251
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.12252    0.07937   1.544 0.133868
## my_trust_value  0.71156    0.15896   4.476 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1028 on 28 degrees of freedom
## Multiple R-squared:  0.4171, Adjusted R-squared:  0.3963
## F-statistic: 20.04 on 1 and 28 DF,  p-value: 0.0001159
##
##
## Call:
## lm(formula = RelSend ~ combine_trust, data = x_combine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28557 -0.06216  0.01940  0.09108  0.18441
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.31285    0.06853   4.566 9.09e-05 ***
## combine_trust  0.67075    0.28048   2.391  0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1227 on 28 degrees of freedom
## Multiple R-squared:  0.1696, Adjusted R-squared:  0.14
## F-statistic: 5.719 on 1 and 28 DF,  p-value: 0.02374
```

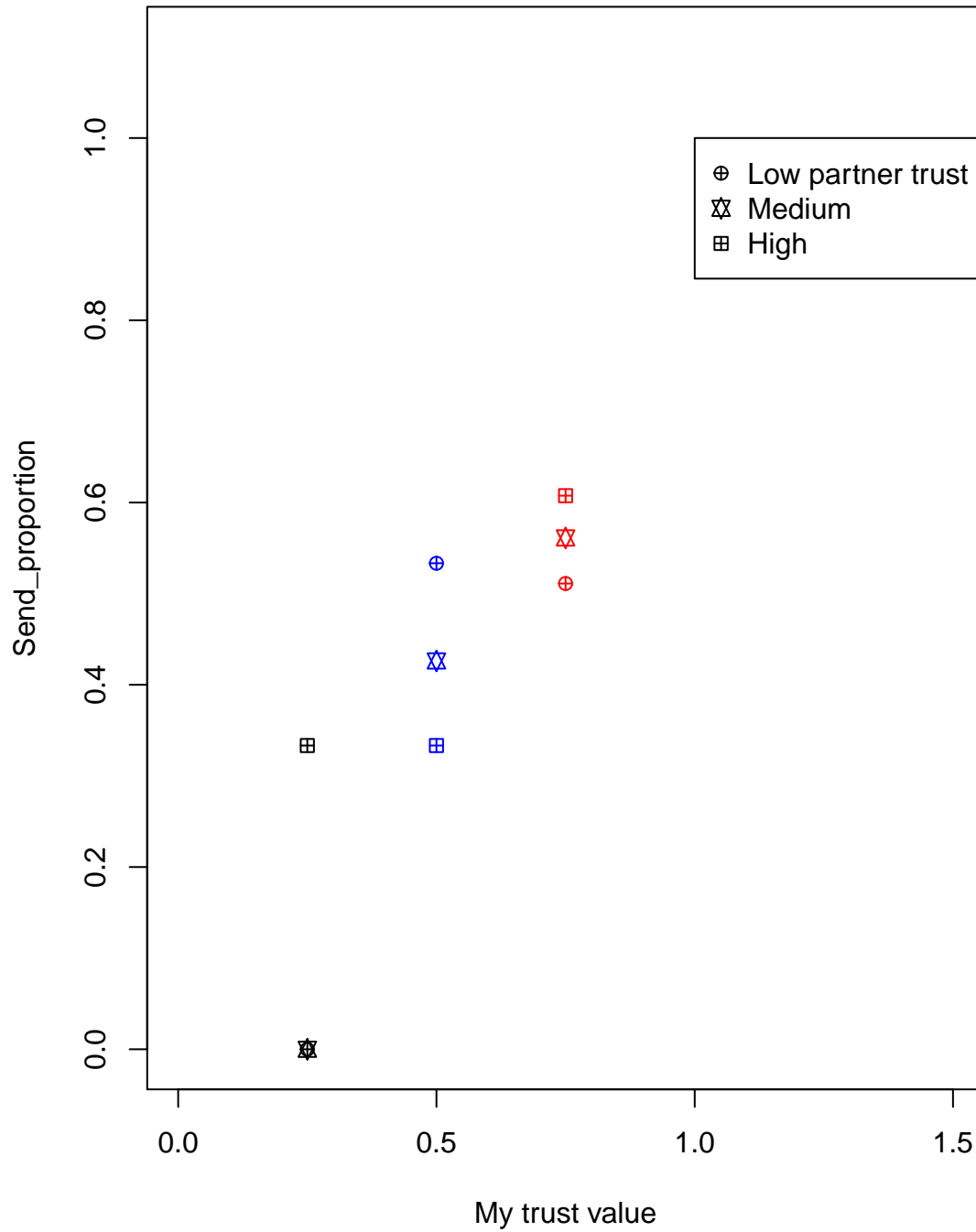
Understanding interactions between two trust values

### Interaction in score game for type SENDER



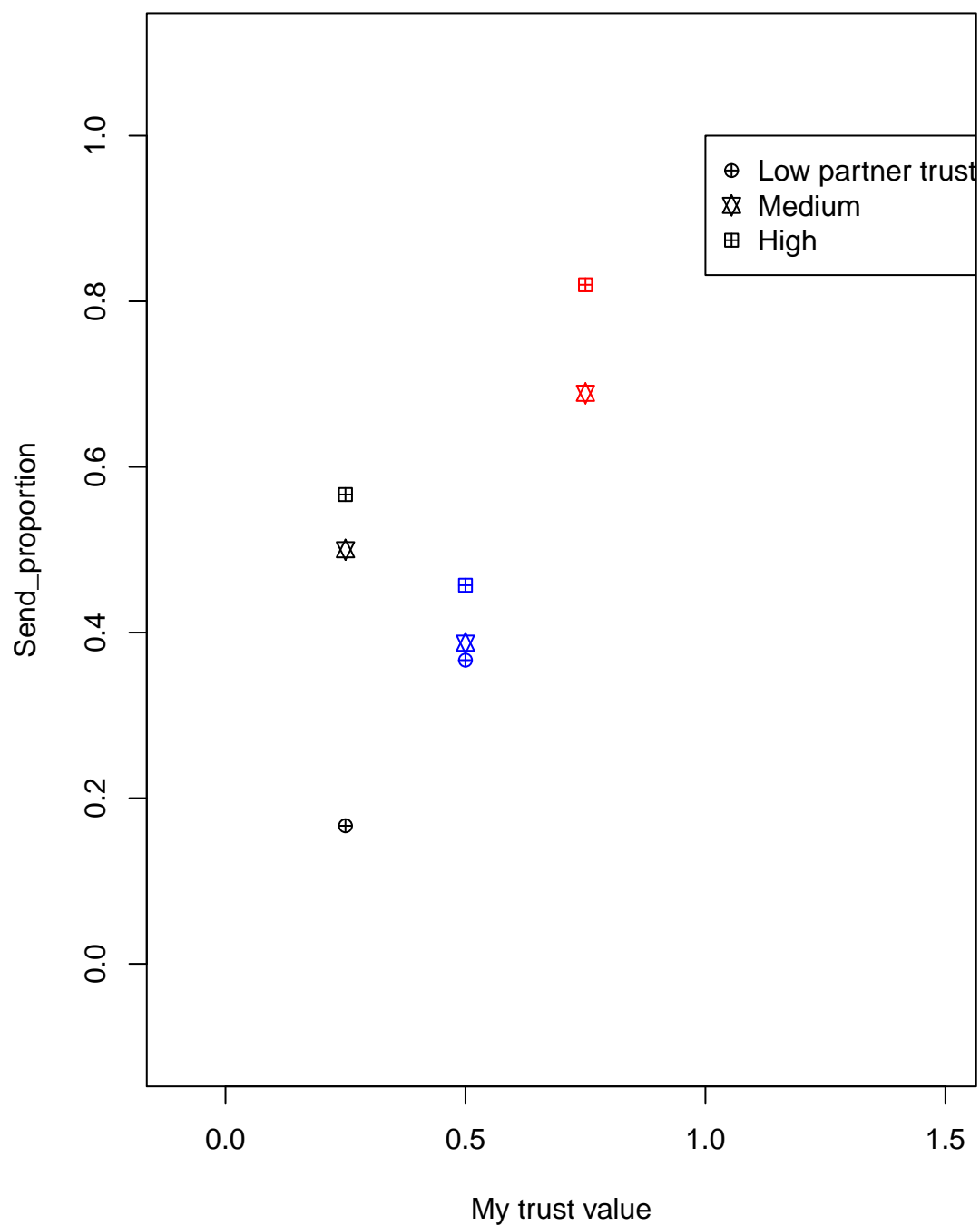
##  
##  
## \pagebreak

## Interaction in score game for type RECEIVER



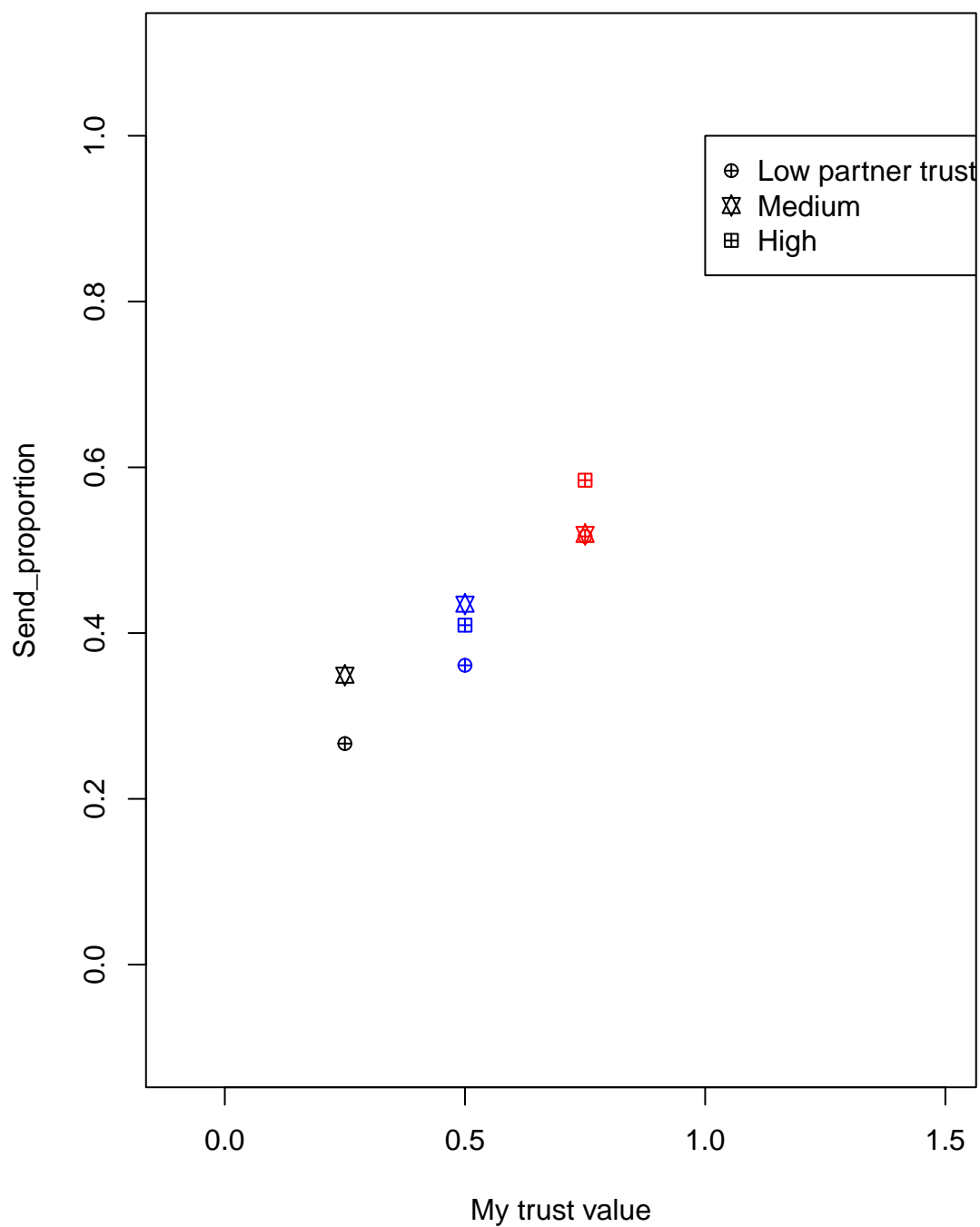
##  
##  
## \pagebreak

### Interaction in combine game for type SENDER



##  
##  
## \pagebreak

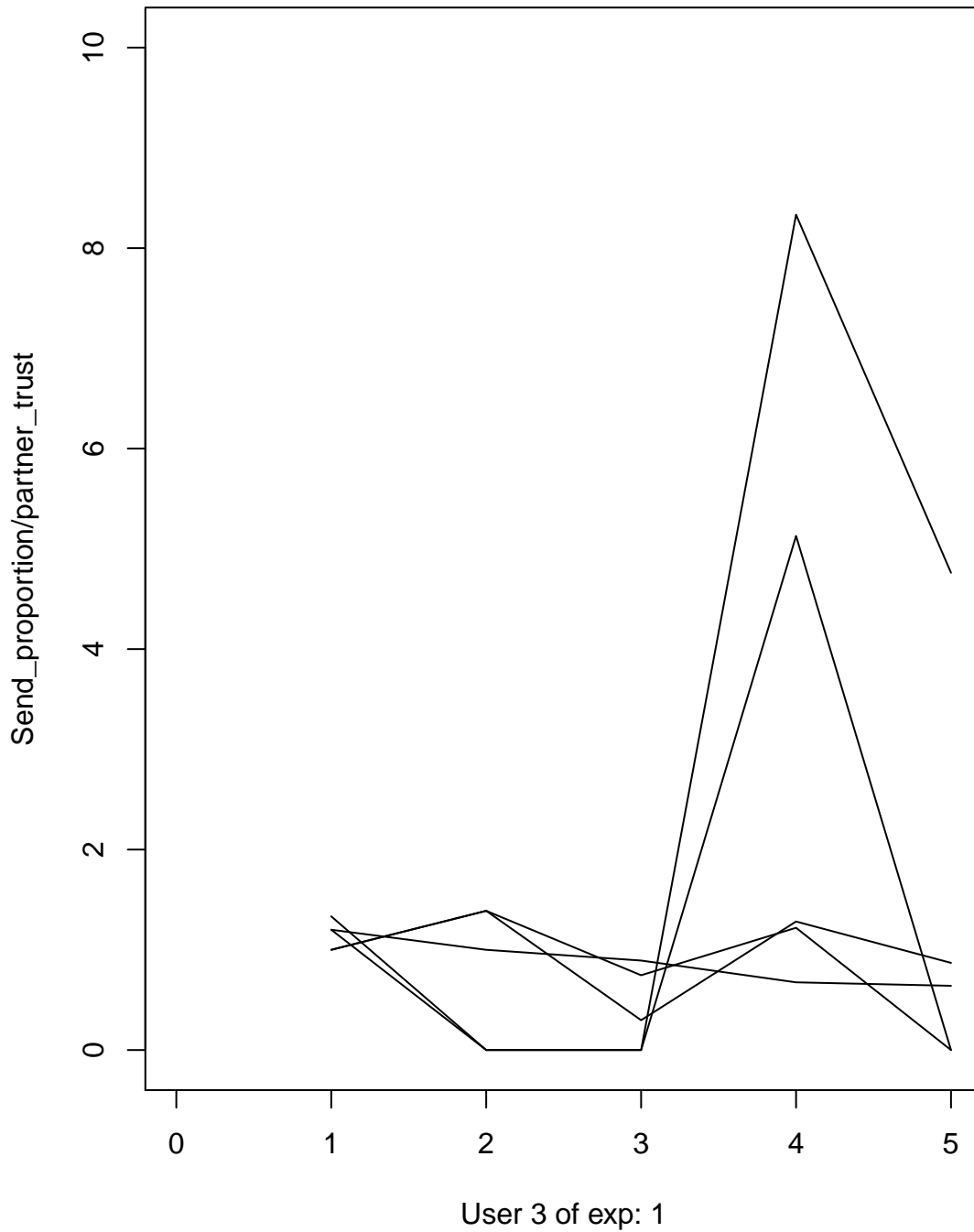
### Interaction in combine game for type RECEIVER



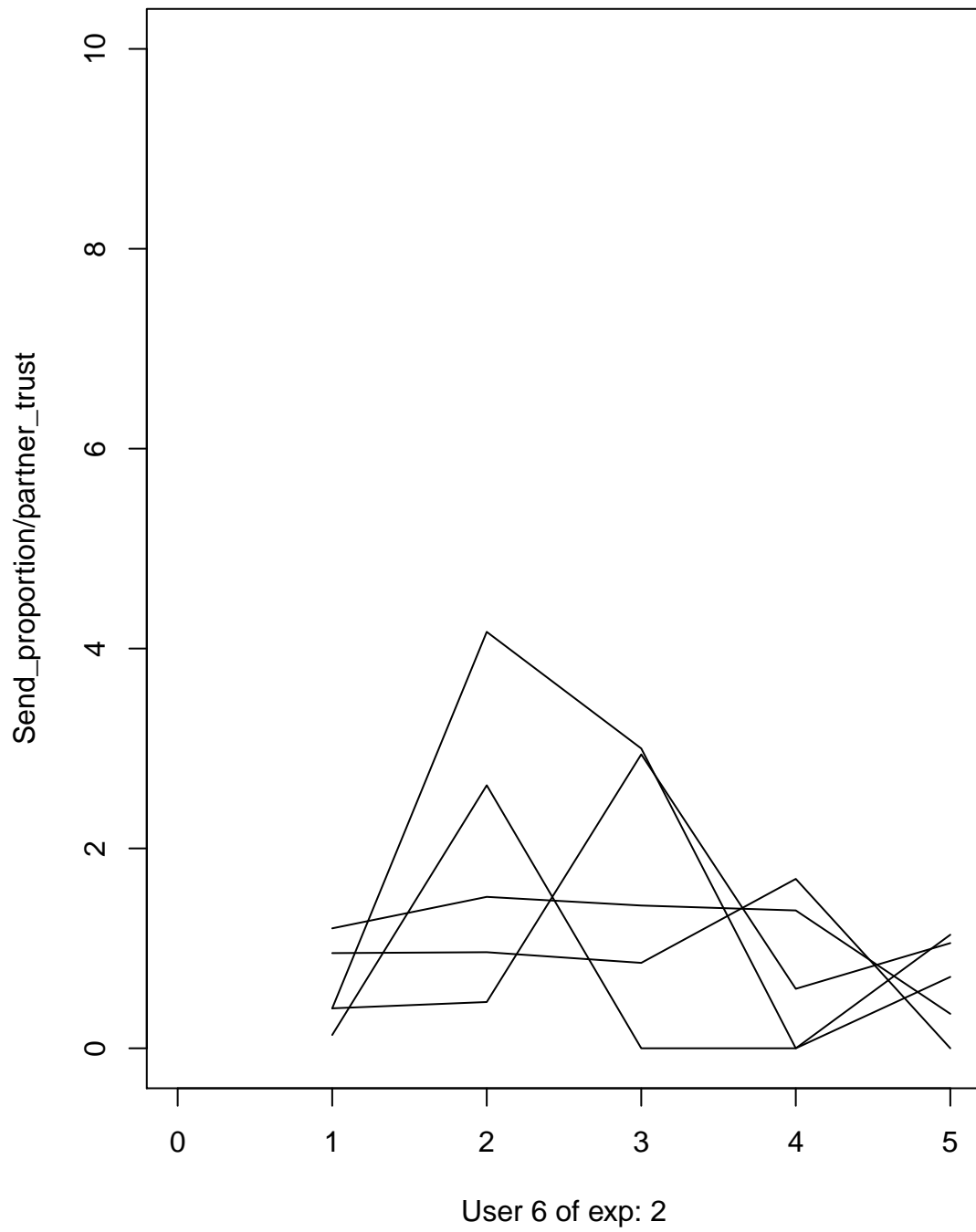
##  
##  
## \pagebreak

## Check the change of user behavior with a partner over time in presence of trust score

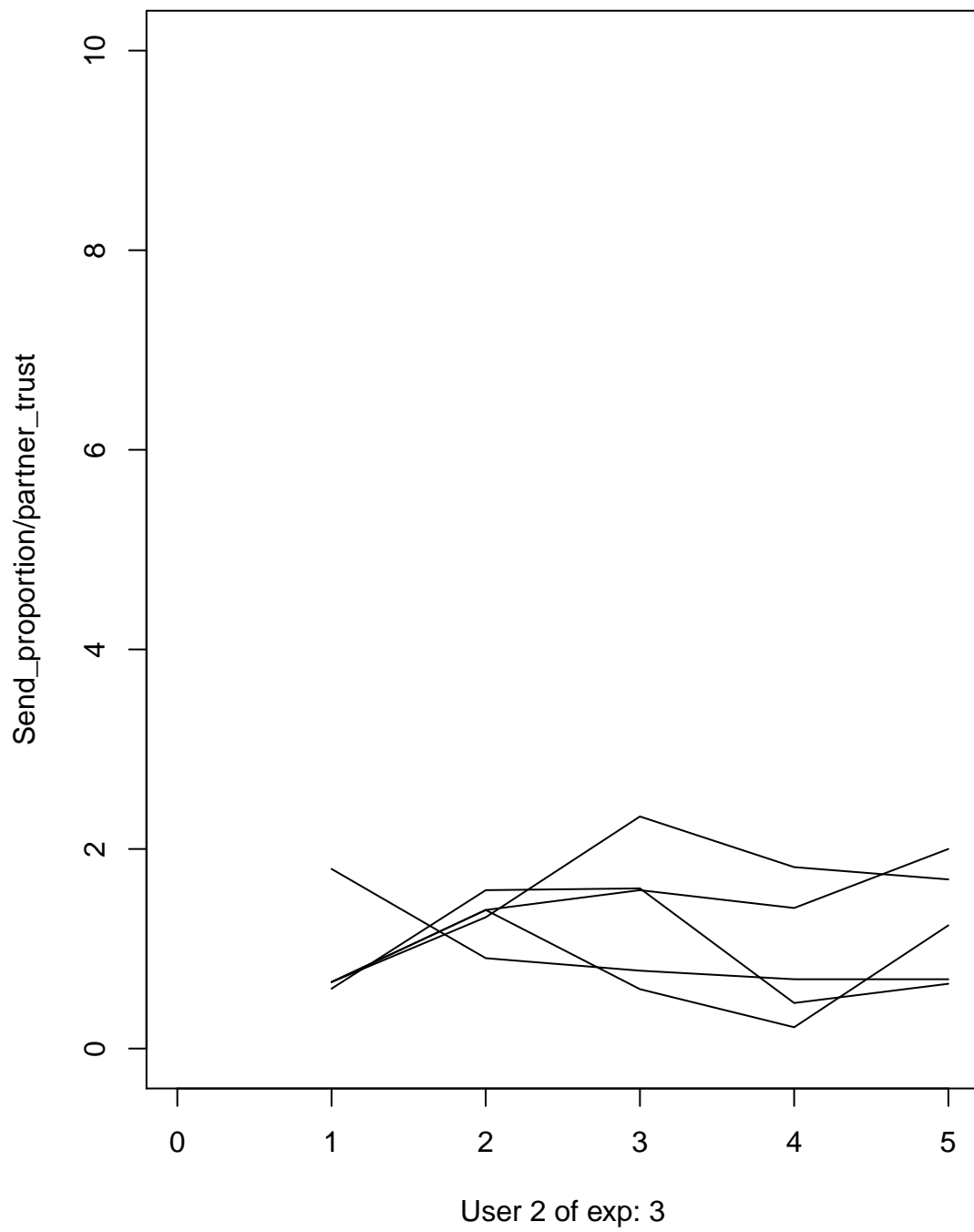
In this section, we will see do the behavior of one user converge in the end of a game.



```
##  
##  
## \pagebreak
```

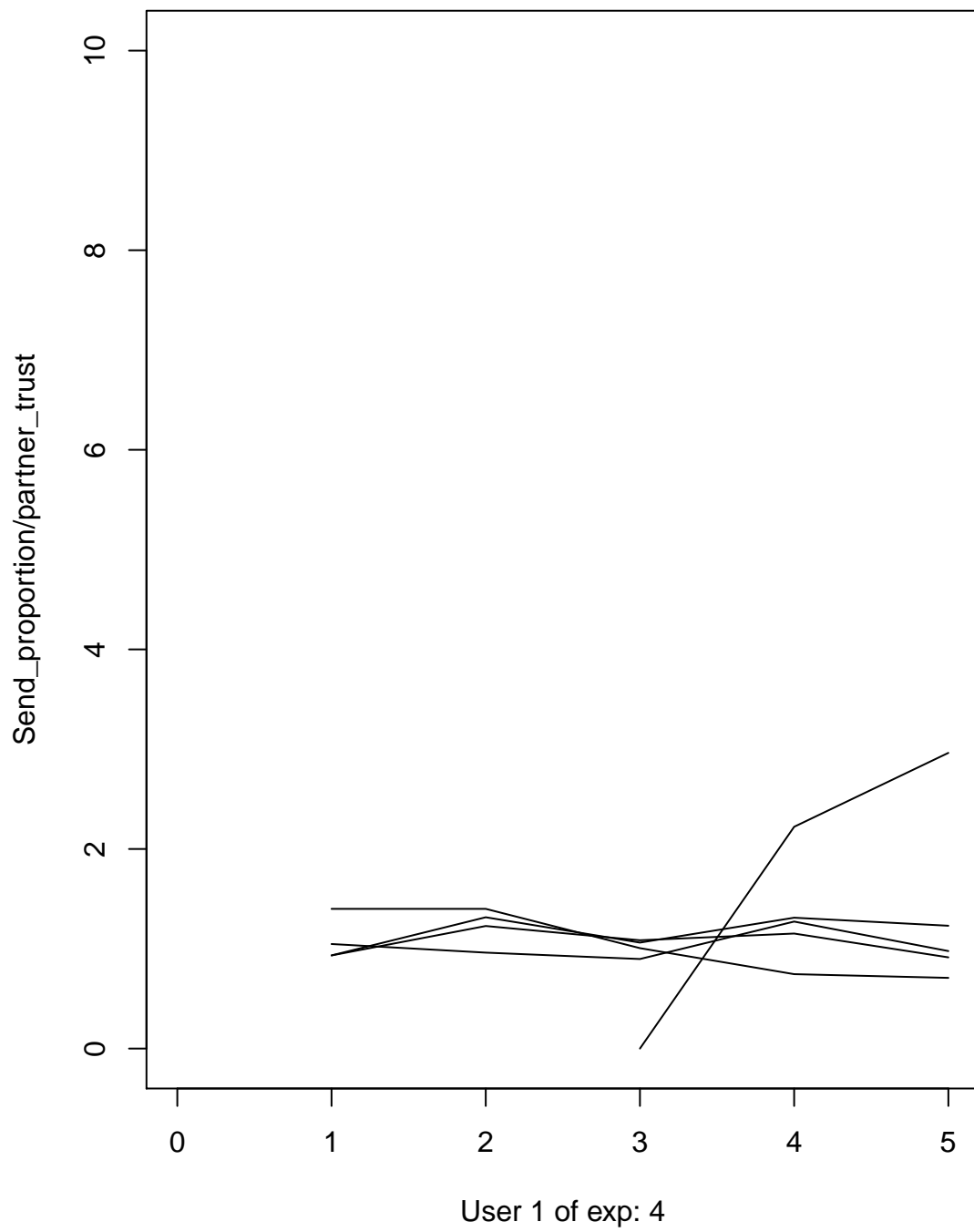


##  
##  
## \pagebreak

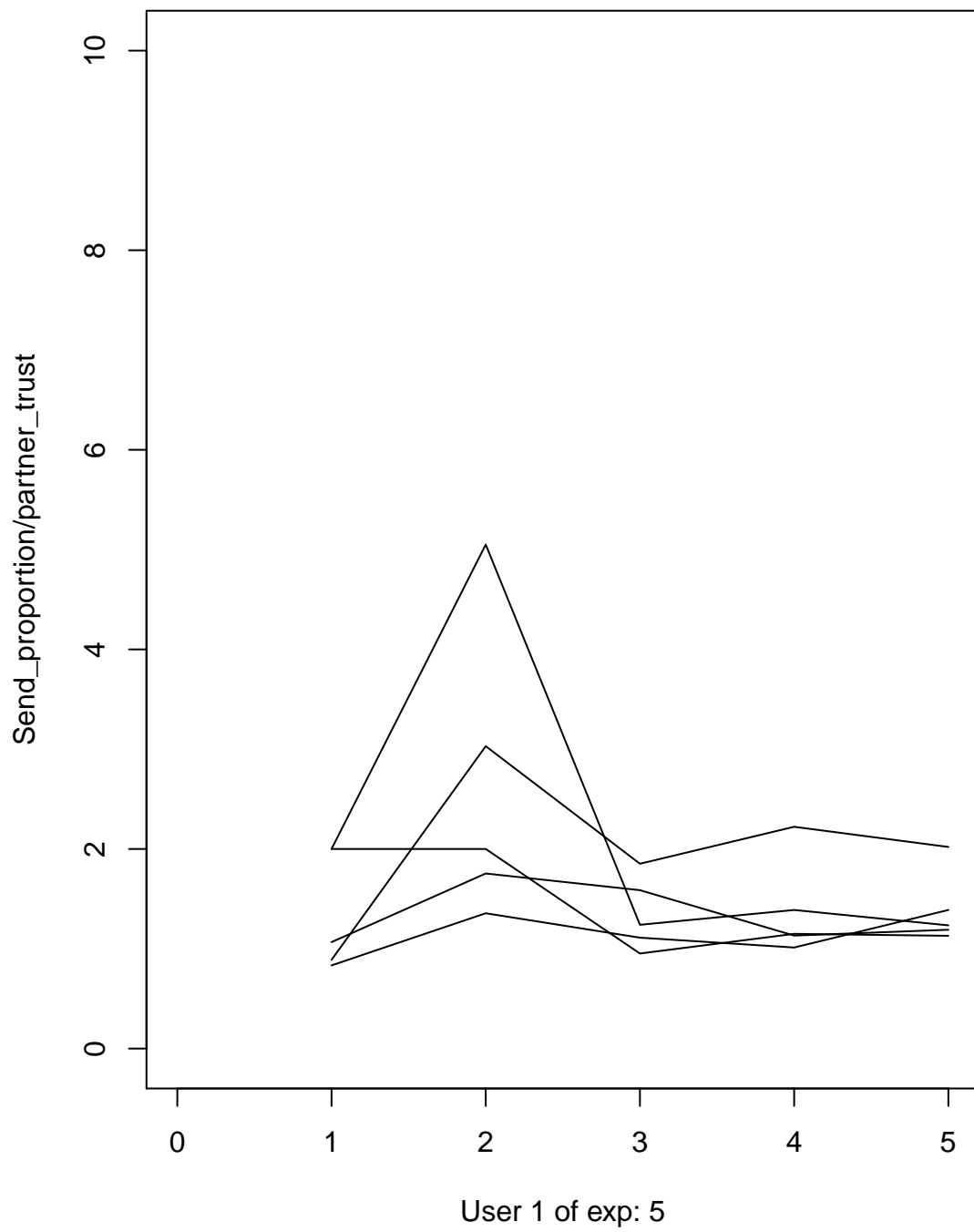


##  
##  
## \pagebreak



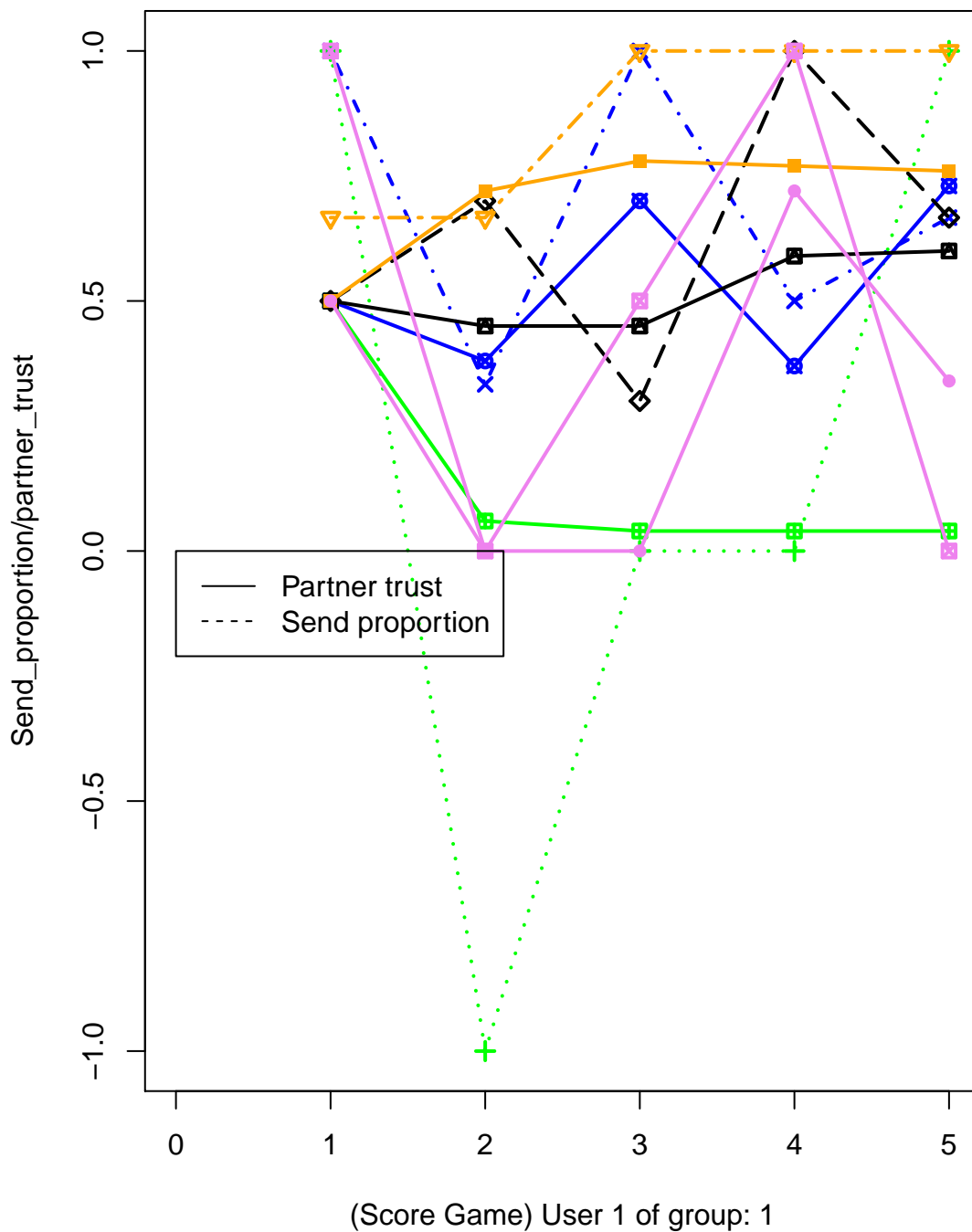


##  
##  
## \pagebreak

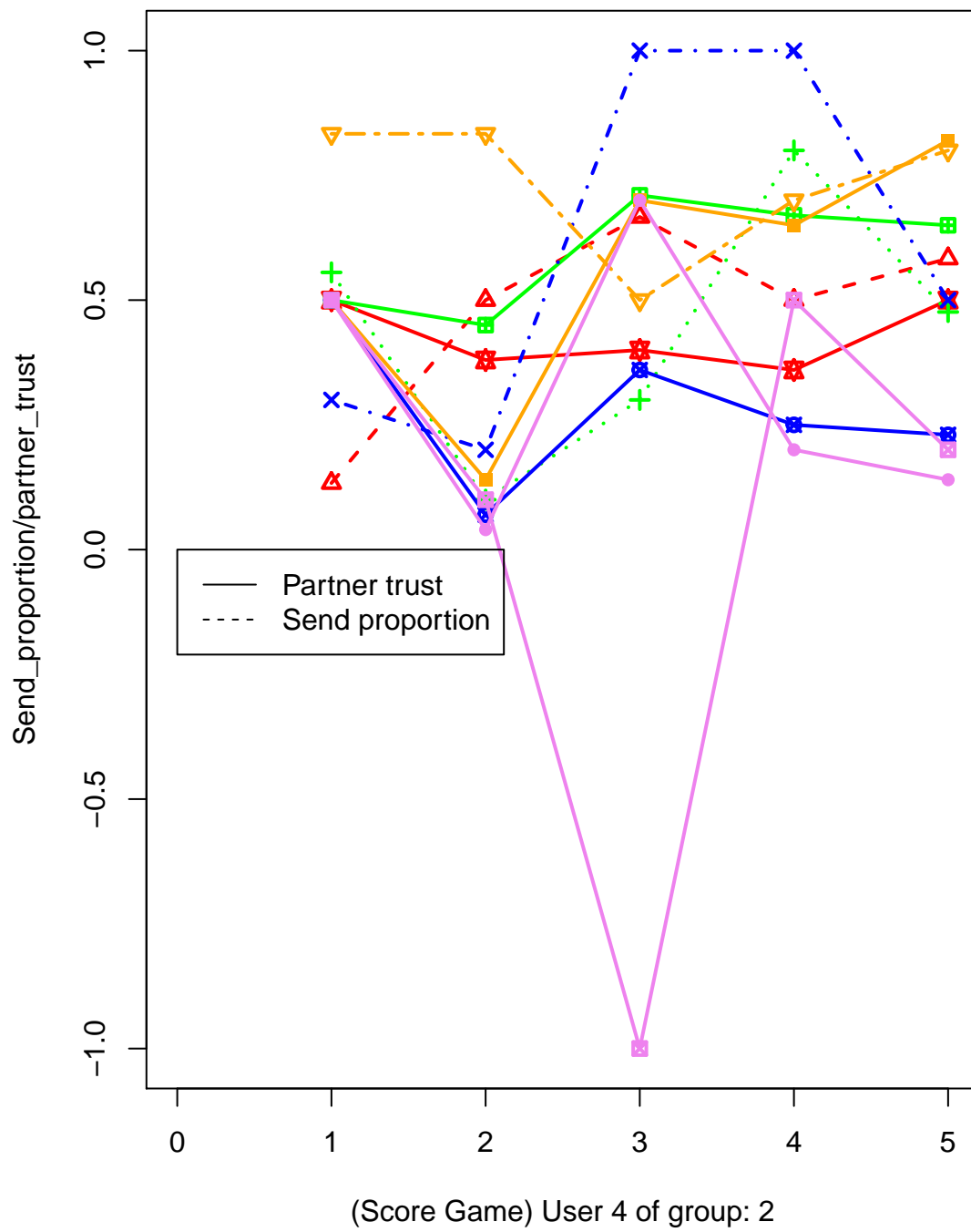


##  
##  
## \pagebreak

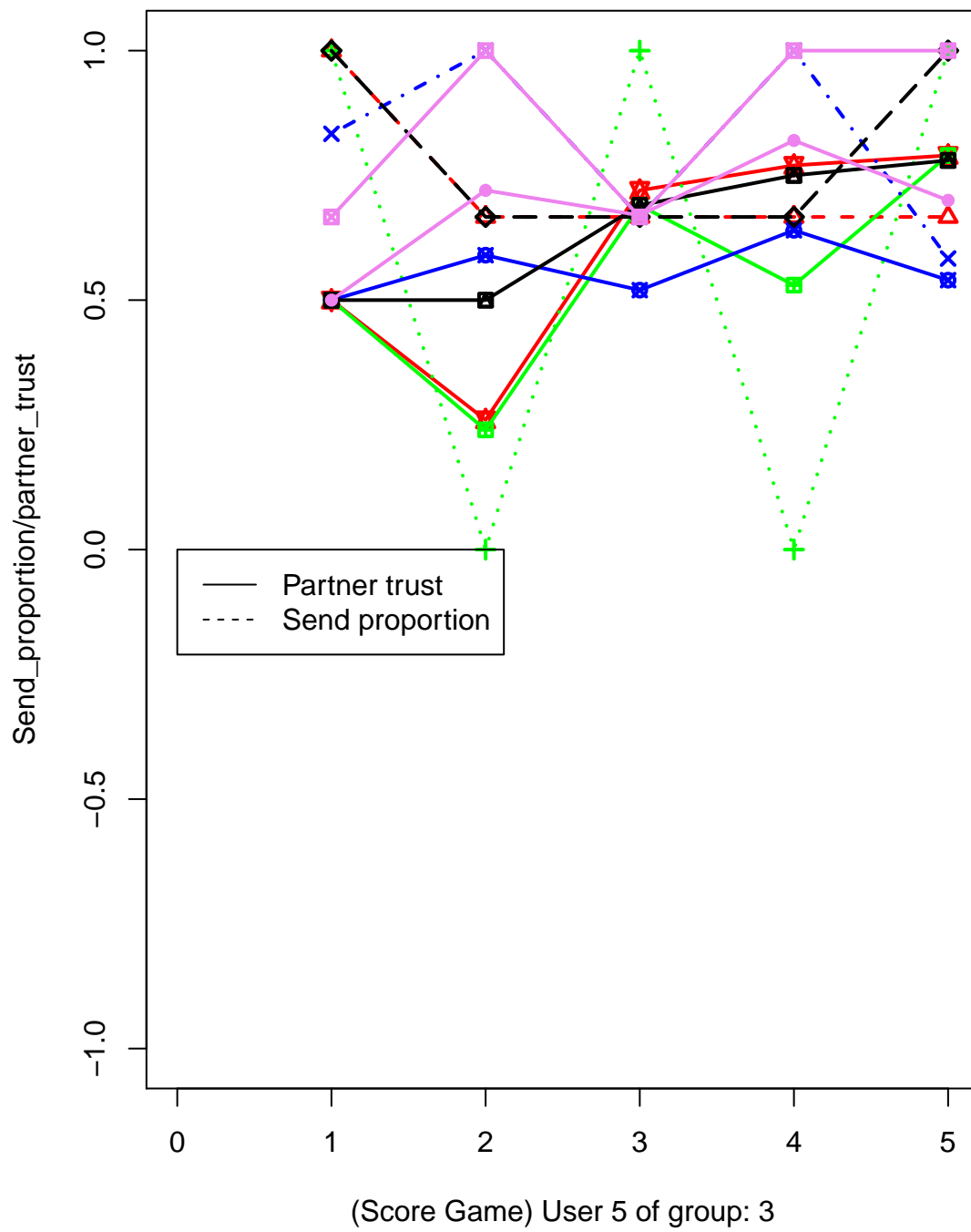
As the metric  $\text{send\_proportion}/\text{partner\_trust}$  does not say anything, we now display both  $\text{send\_proportion}$  and  $\text{partner\_trust}$  in one graph to check the stability of user behavior over time.



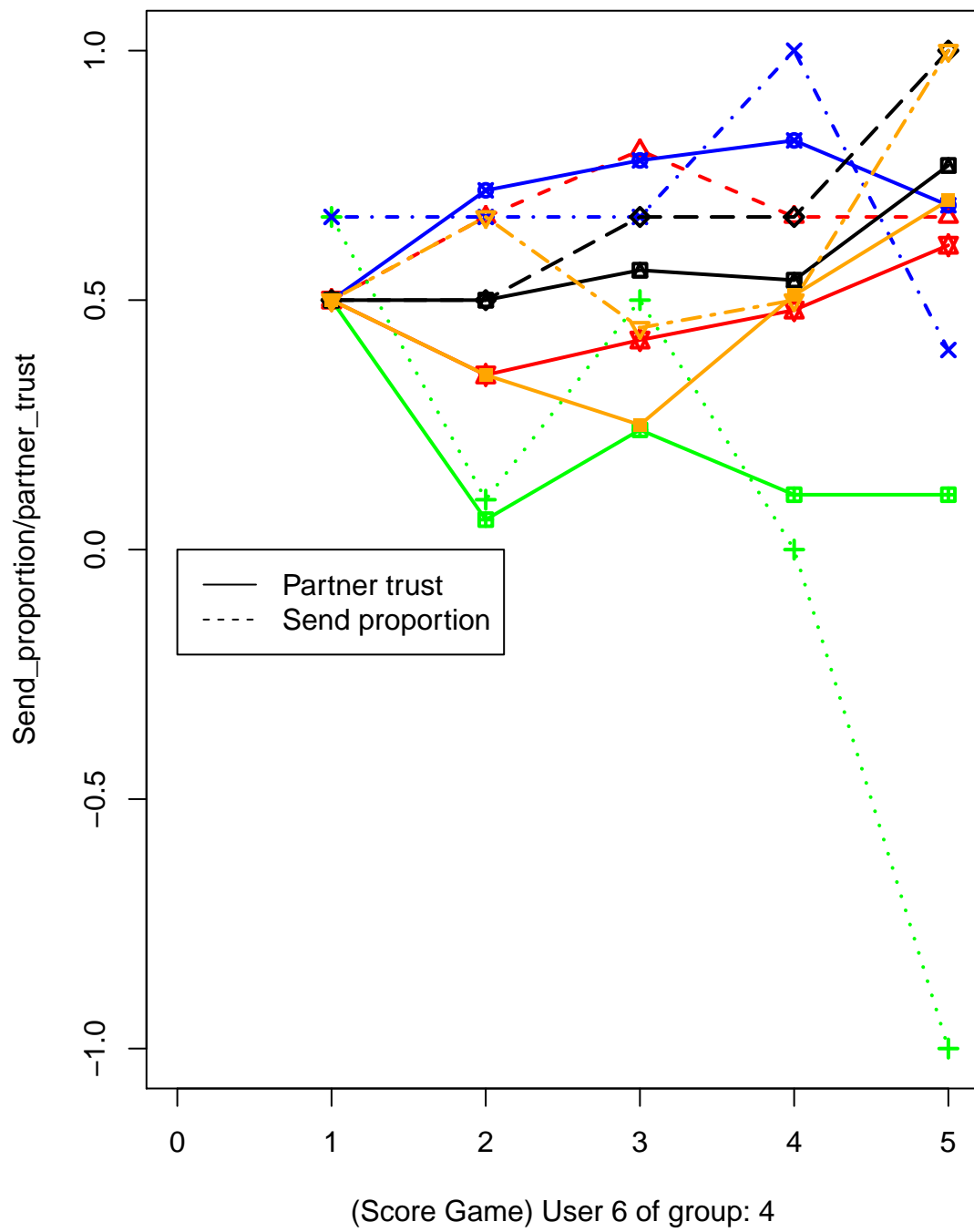
##  
##  
## \pagebreak



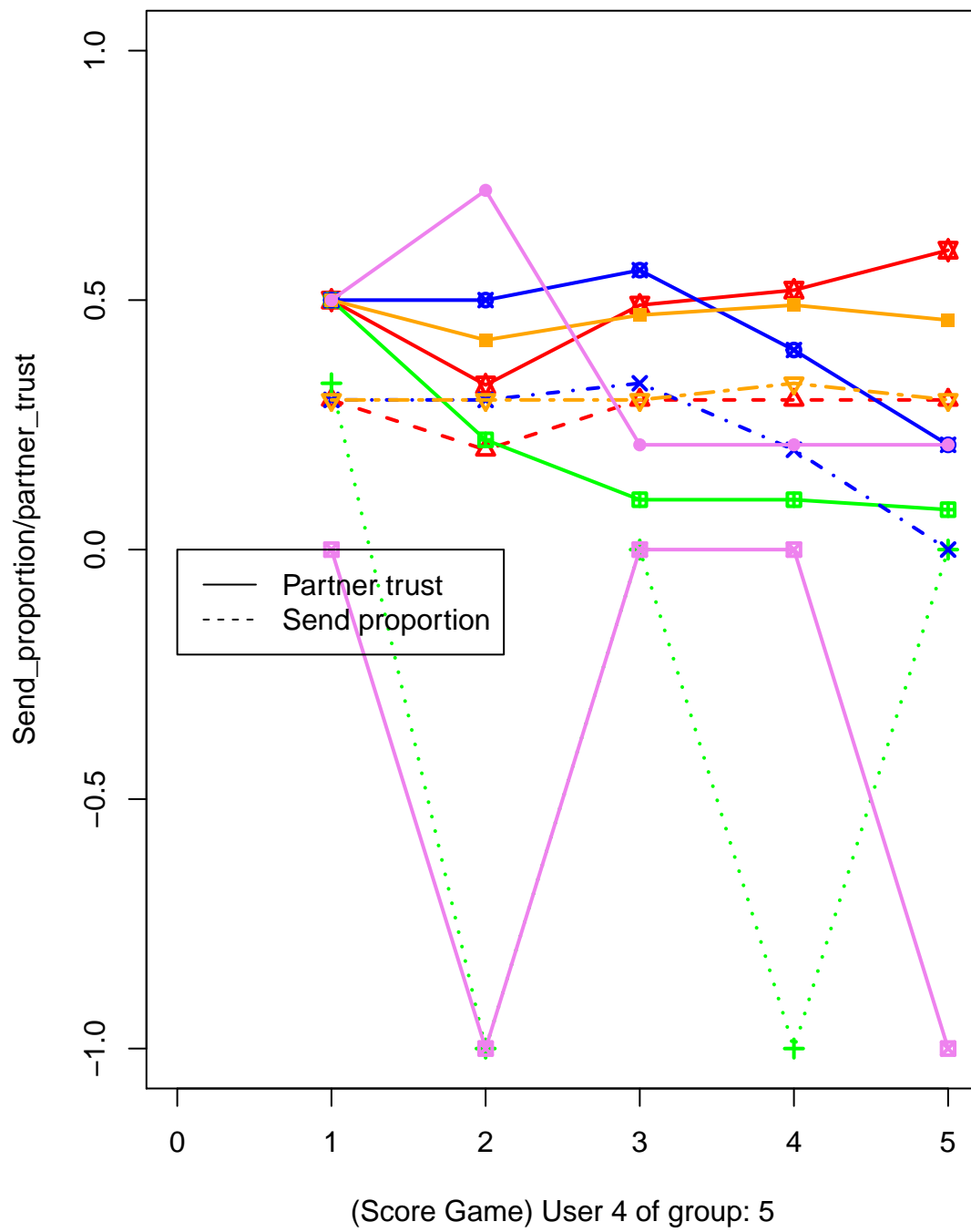
##  
##  
## \pagebreak



##  
##  
## \pagebreak



##  
##  
## \pagebreak

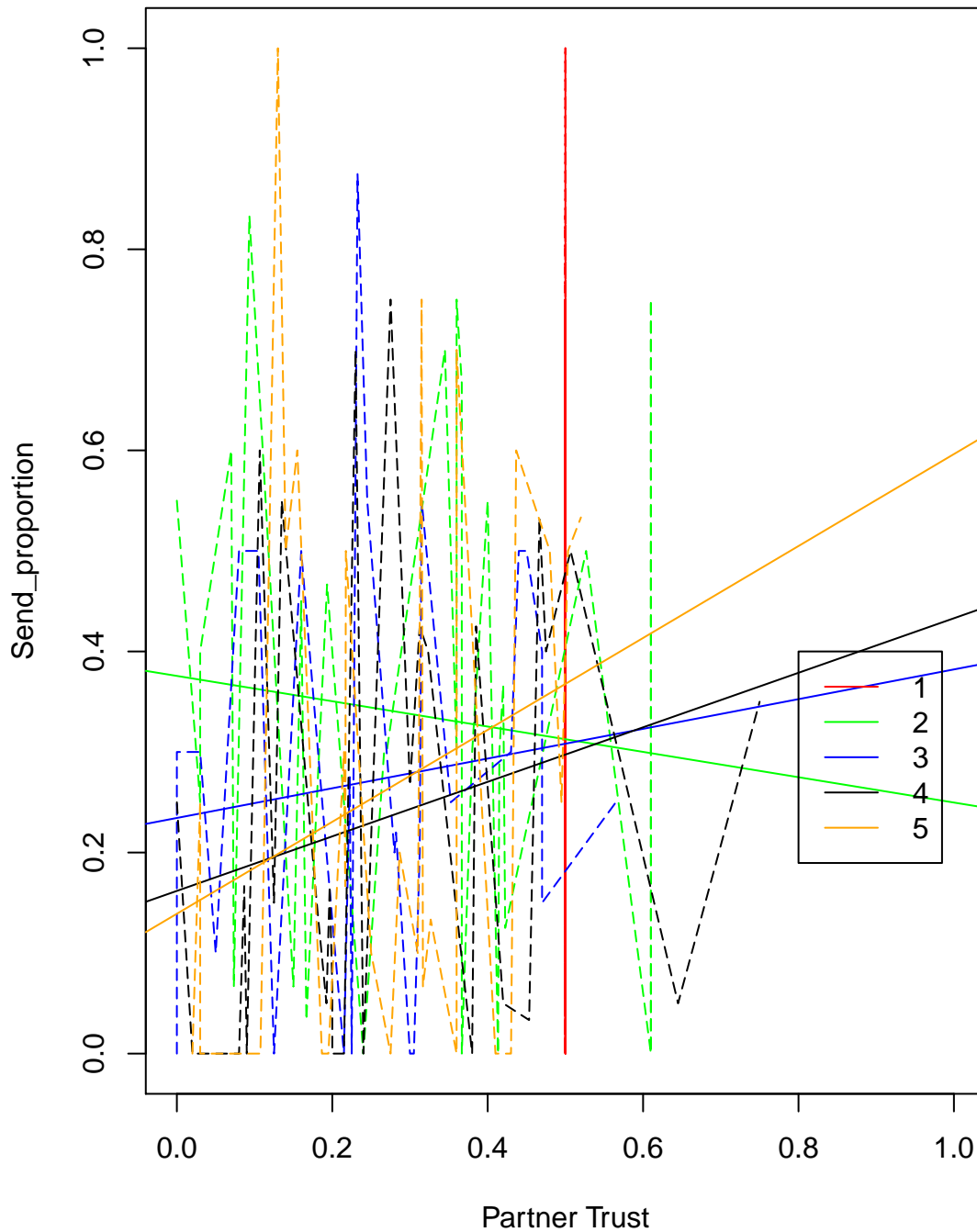


##  
##  
## \pagebreak

Change of send\_proportion ~ partner\_trust over time

We want to see how user send proportion depends on partner trust in each round for each game

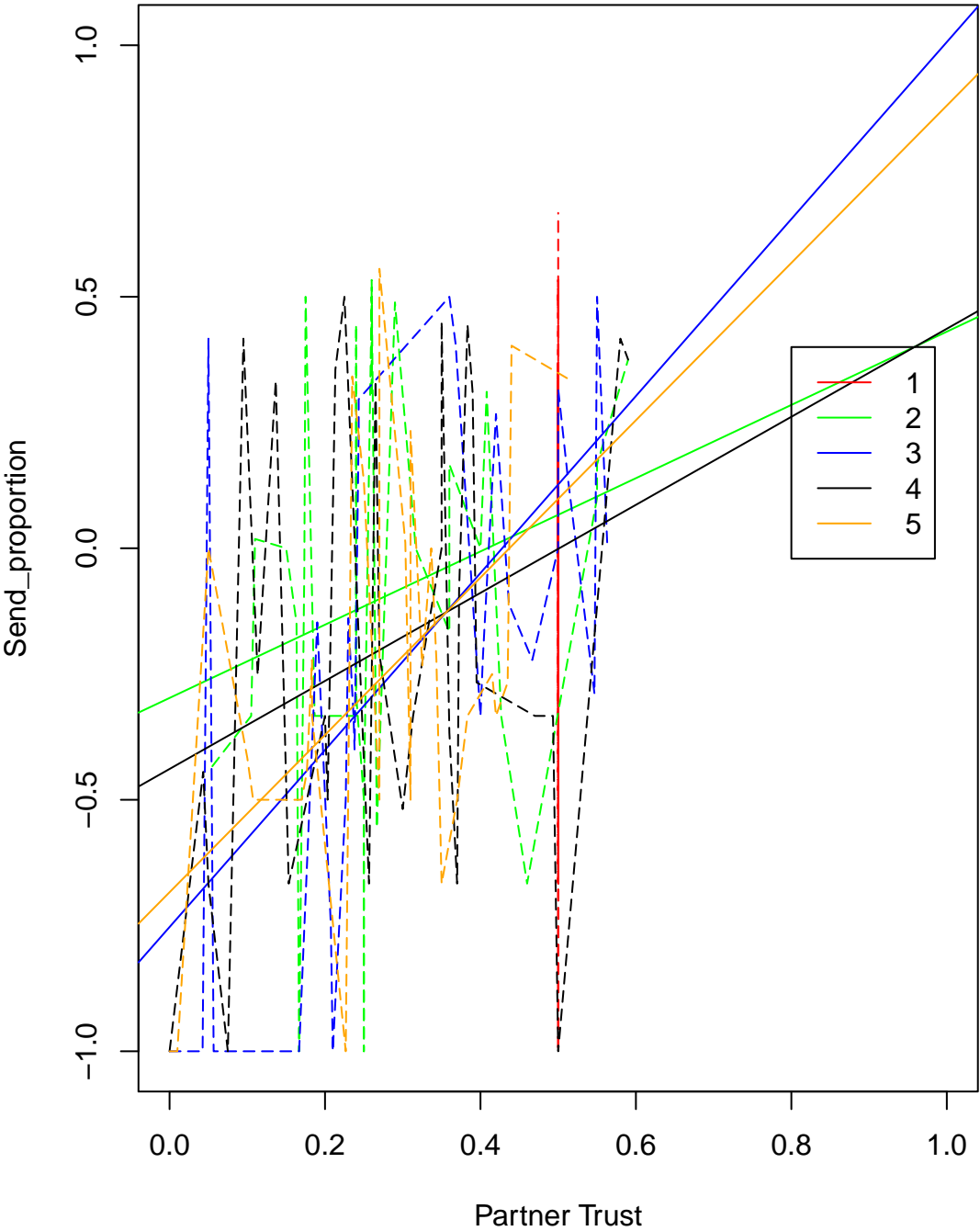
### Simple Game SENDER



```
##  
##  
## \pagebreak
```

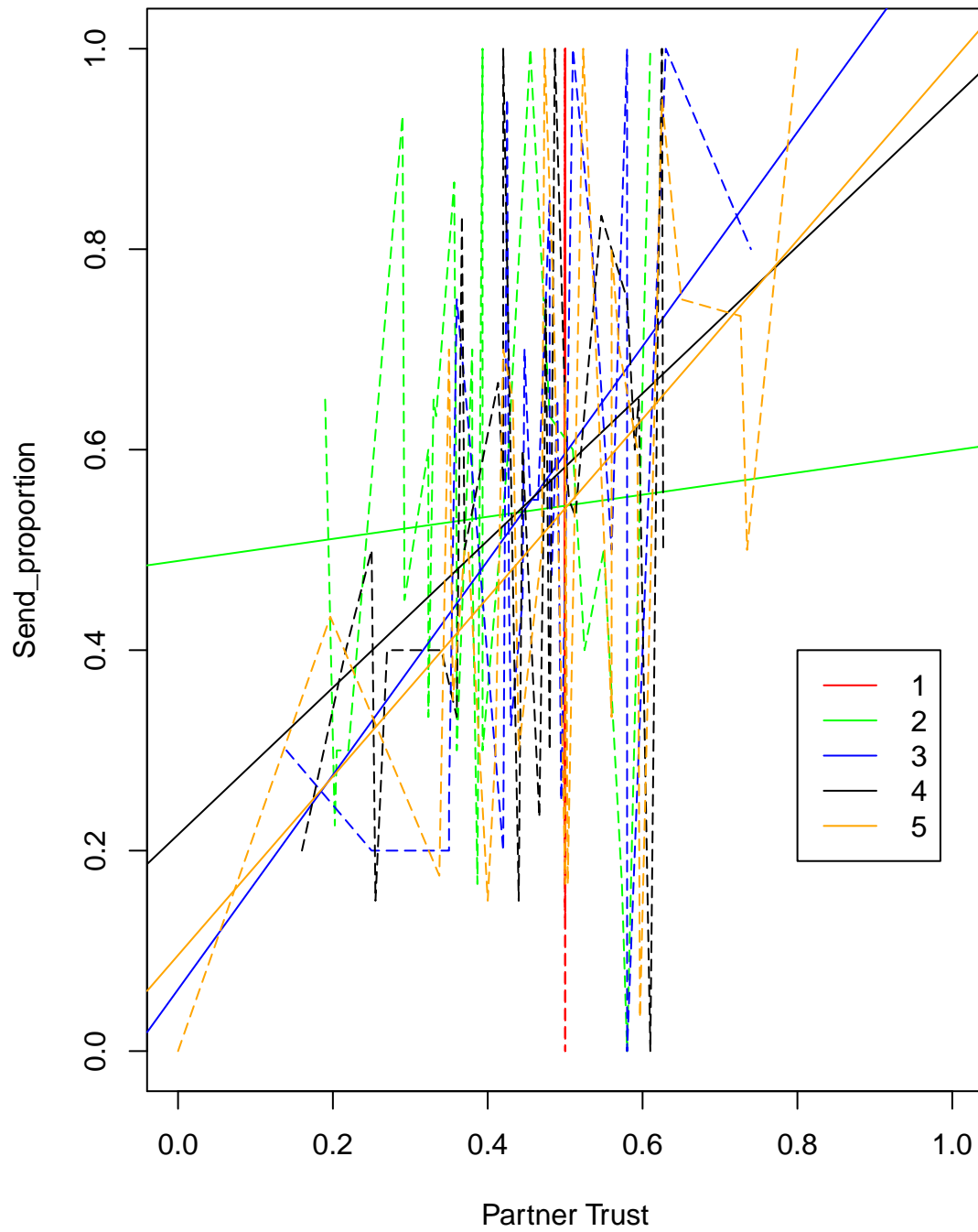


Simple Game RECEIVER



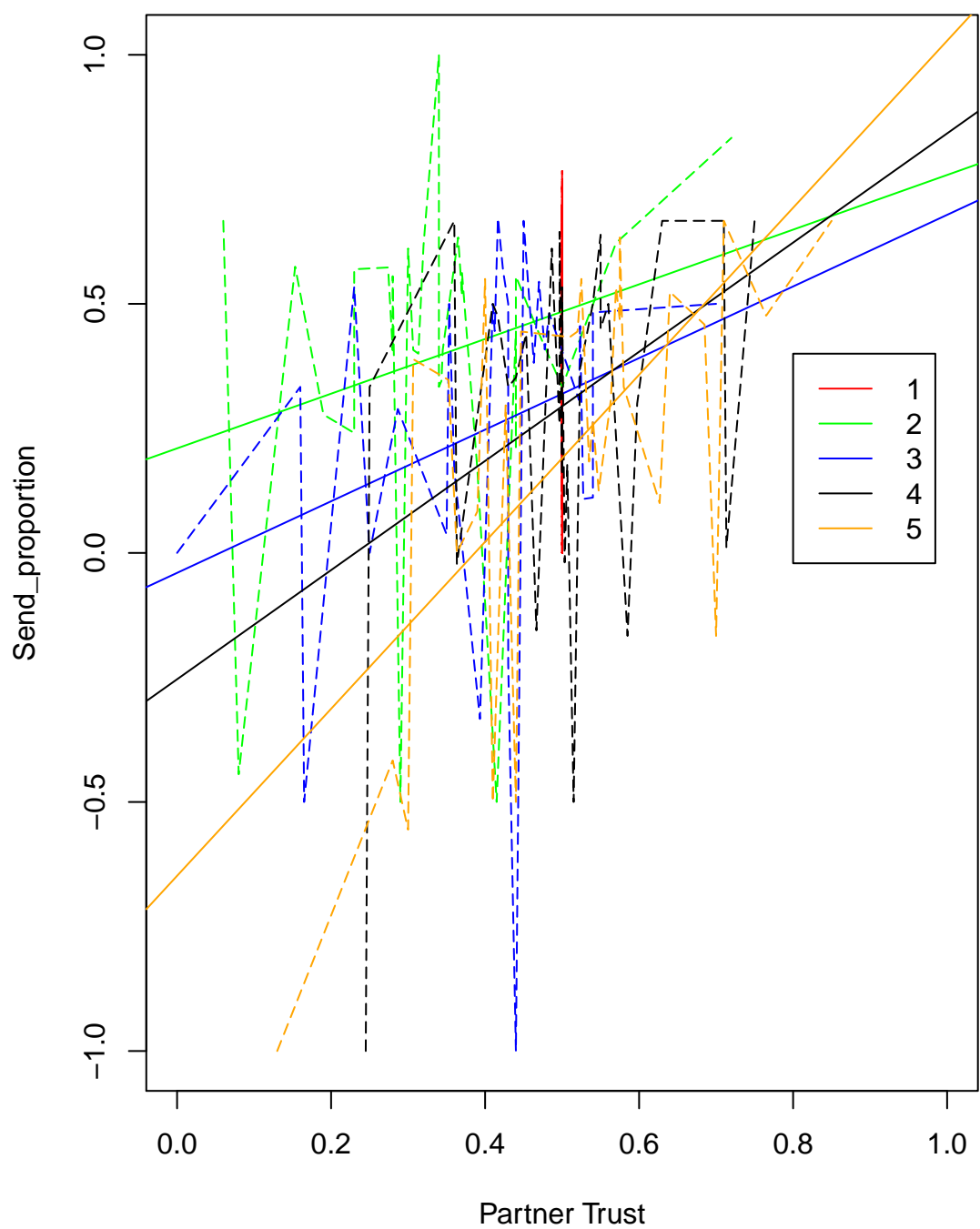
##  
##  
## \pagebreak

# ID Game SENDER



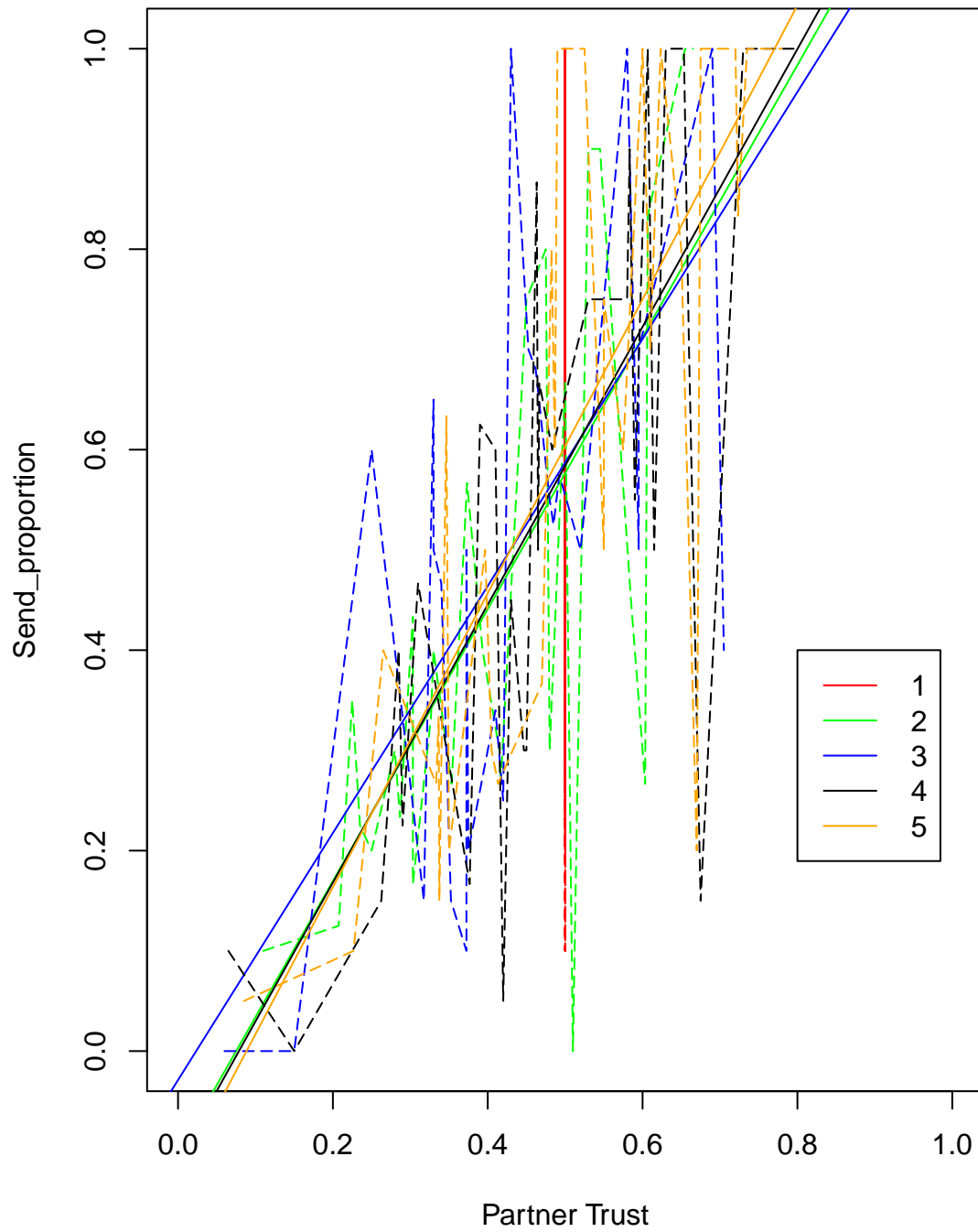
##  
##  
## \pagebreak

# ID Game RECEIVER



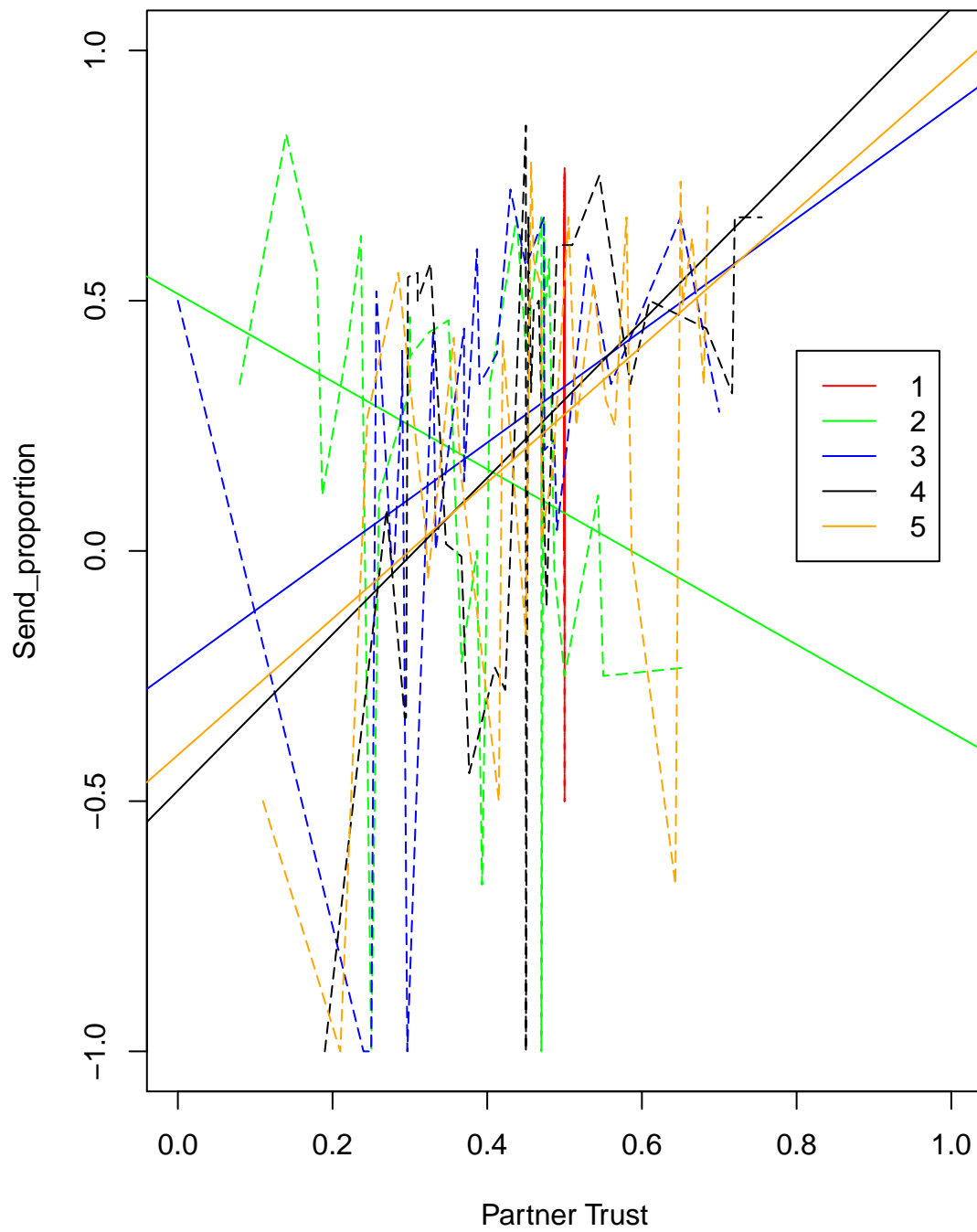
##  
##  
## \pagebreak

# Score Game SENDER



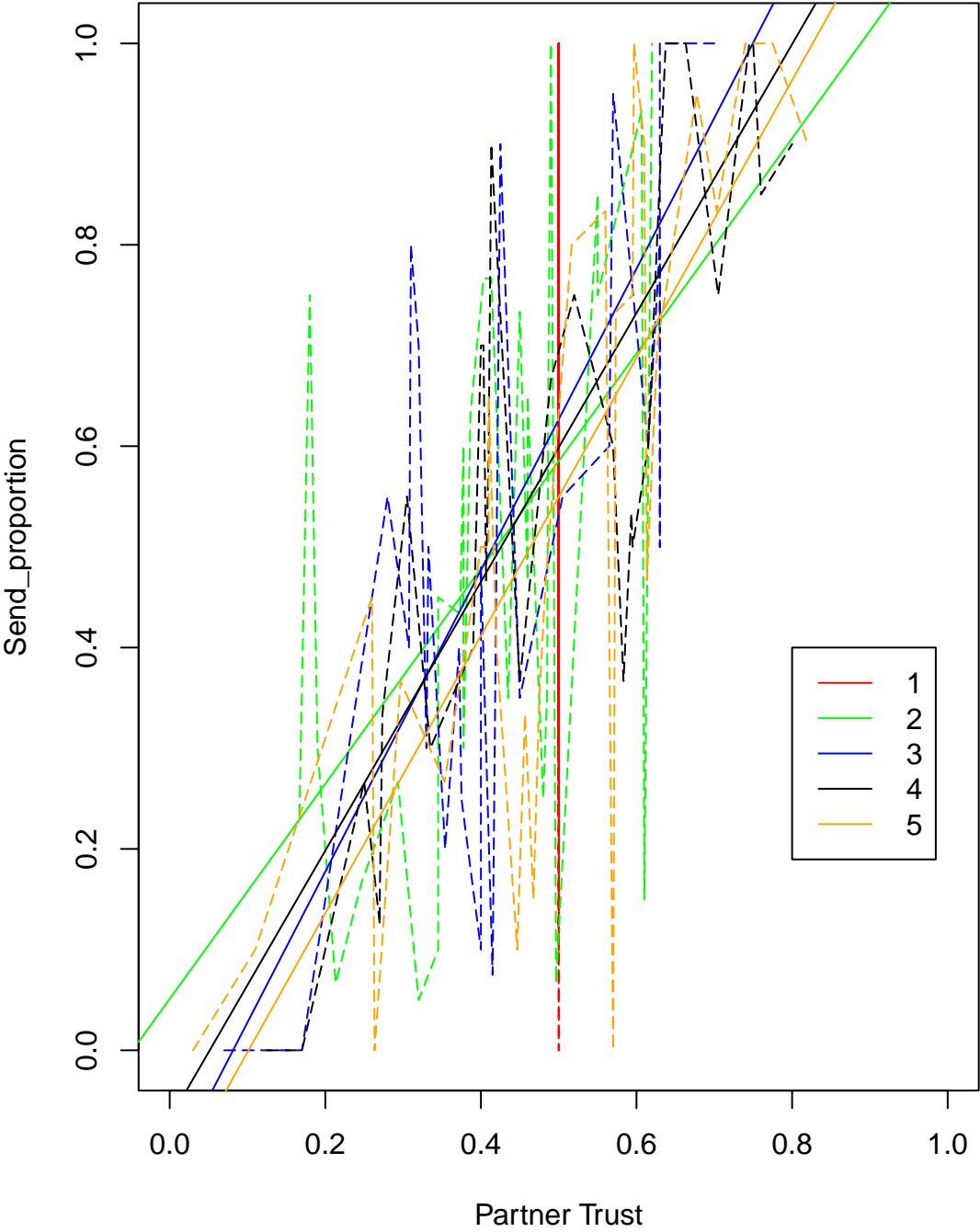
##  
##  
## \pagebreak

## Score Game RECEIVER



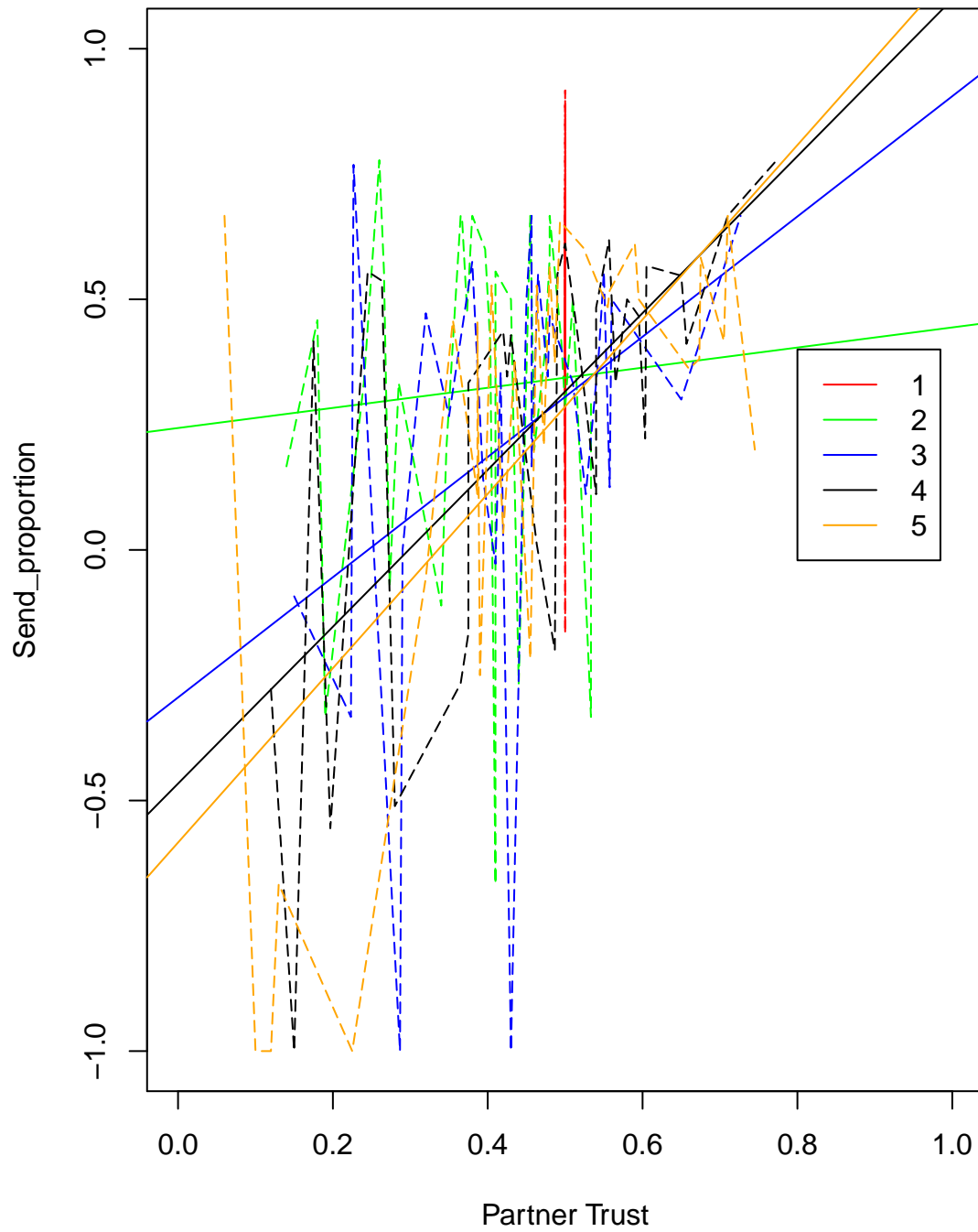
##  
##  
## \pagebreak

Combine Game SENDER



##  
##  
## \pagebreak

## Combine Game RECEIVER



```
##  
##  
## \pagebreak
```

## Group Effect on Regression Power (Sending ~ 2 trust scores)



## **Analyze the group effect on regression of sending behavior and own trust value**

Also, we want to see can we predict the behavior of users based on trust score.

Temporary disable, because it is better to regression on 2 trust scores instead of 1

## **Analyze the group effect on correlation of sending behavior and own trust value**

We want to see if the correlation (prediction power) of trust score we calculated to users and his own future action are consistency between groups.

Temporary disable because regression on 2 trust scores is better.

### **Analyze the group effect on correlation of sending behavior and own trust value without Group 3**

We want to see if the correlation (prediction power) of trust score we calculated to users and his own future action are consistency between groups. From the above analysis, we can see that Group 3 is somehow strange, so we want to analyze without them.

Temporary remove

## **Analyze the correlation of behavior on the trust score of partners showed to users**

We want to analyze the difference between groups, about the correlation of sending behavior and trust value of the partners.

Temporary disable because regression on 2 trust scores is better.

Analyze the correlation of behavior on the trust score of partners showed to users without Group 3

Temporary remove

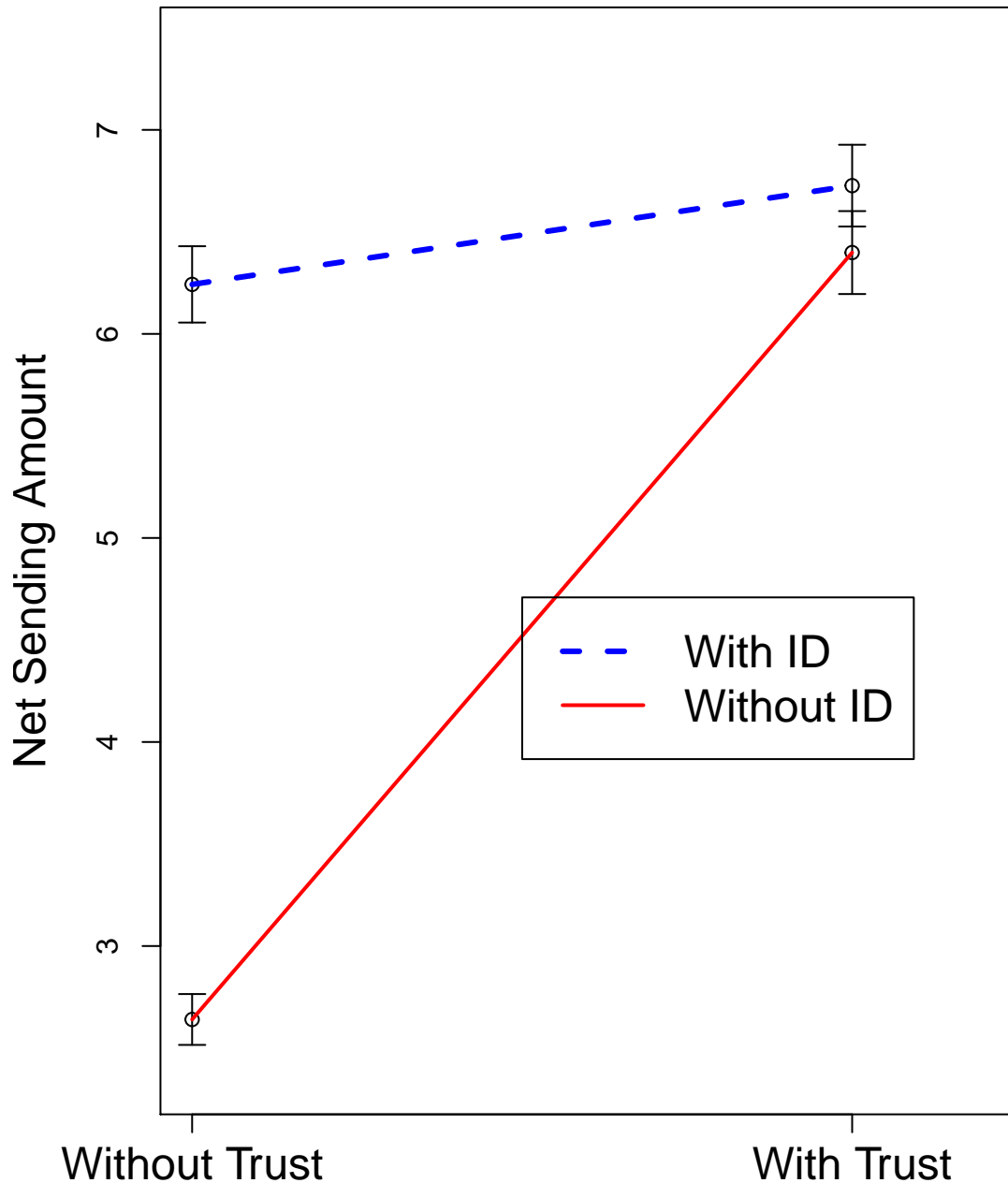
## Analyze on game and group and Tukey test

We analyze the difference between each group for each game

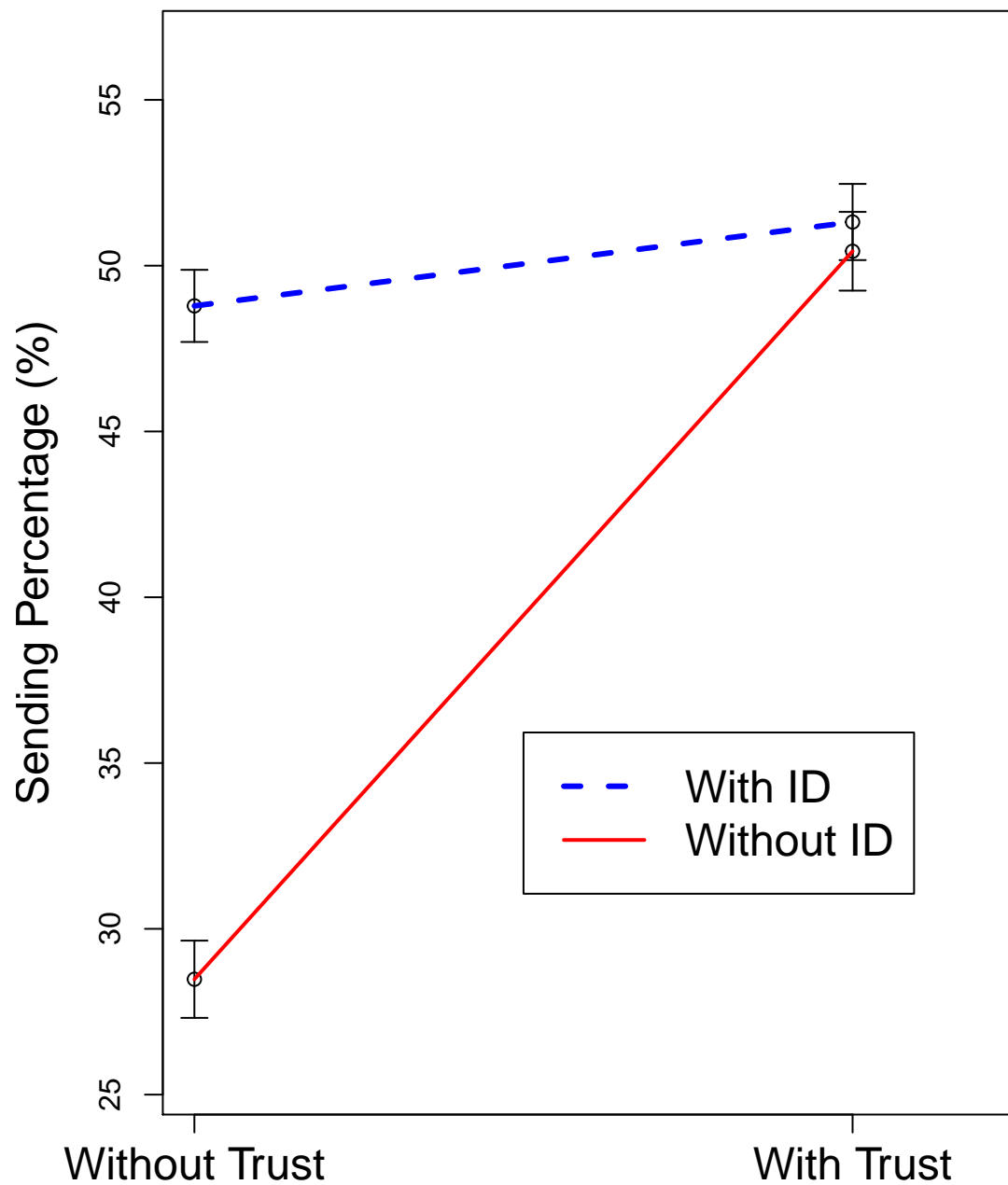
## Plotting data

We plot absolute sending, relative sending and profit for each games, using standard error bars display.

```
plotGameMetrics(simple_games$Contribution, id_games$Contribution,  
  score_games$Contribution, combine_games$Contribution,  
  metric_name = "Net Sending Amount")
```

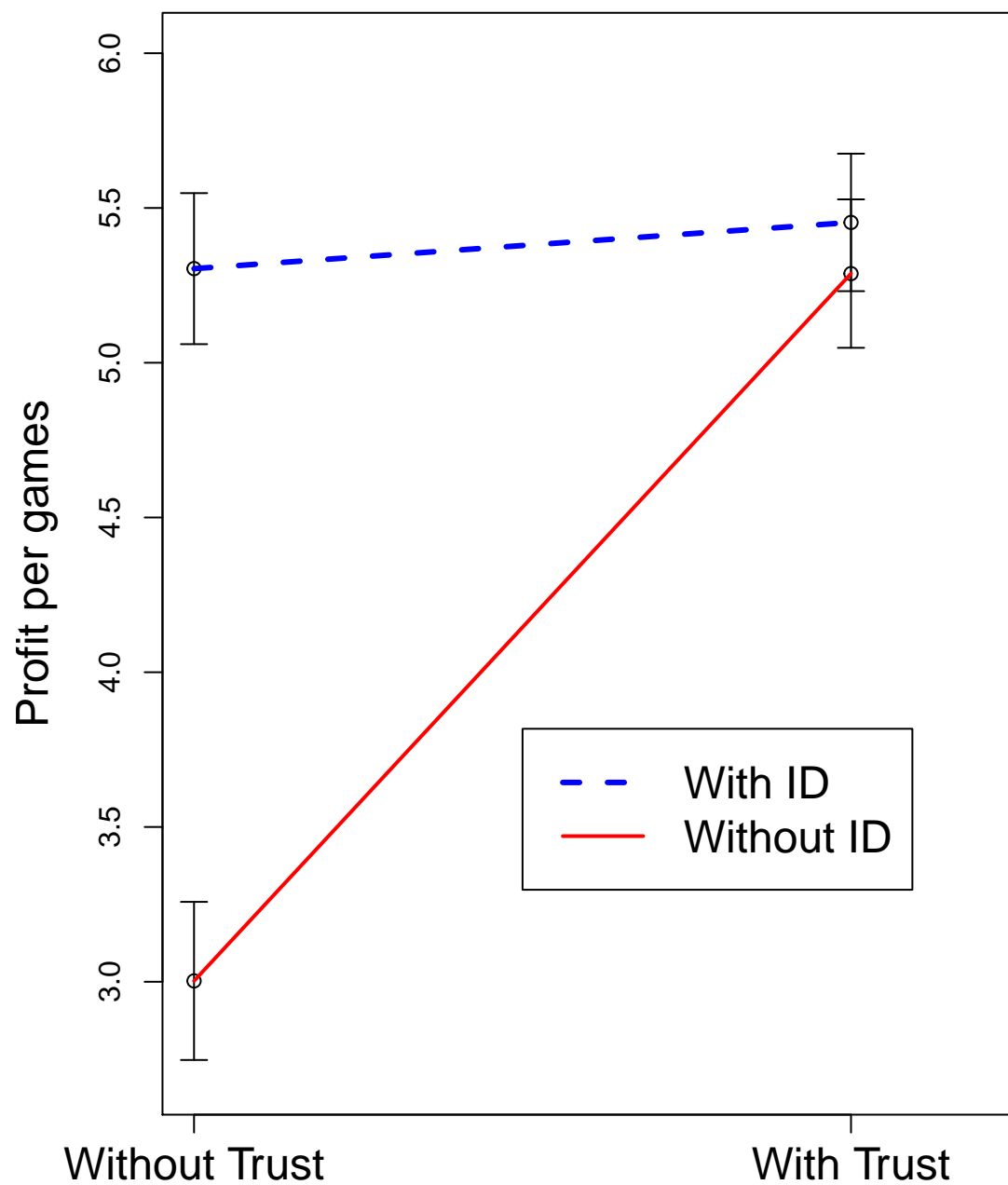


```
plotGameMetrics(simple_games[simple_games$my_send_proportional >= 0, ], $my_send_proportional * 100, id_games[id_games$my_send_proportional >= 0, ], $my_send_proportional * 100, score_games[score_games$my_send_proportional >= 0, ], $my_send_proportional * 100, combine_games[combine_games$my_send_proportional >= 0, ], $my_send_proportional * 100, metric_name = "Sending Percentage (%)")
```



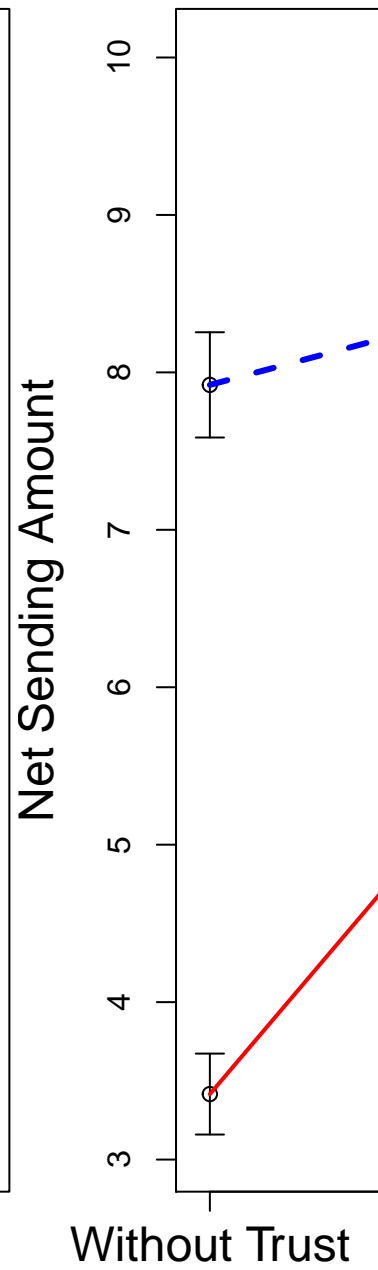
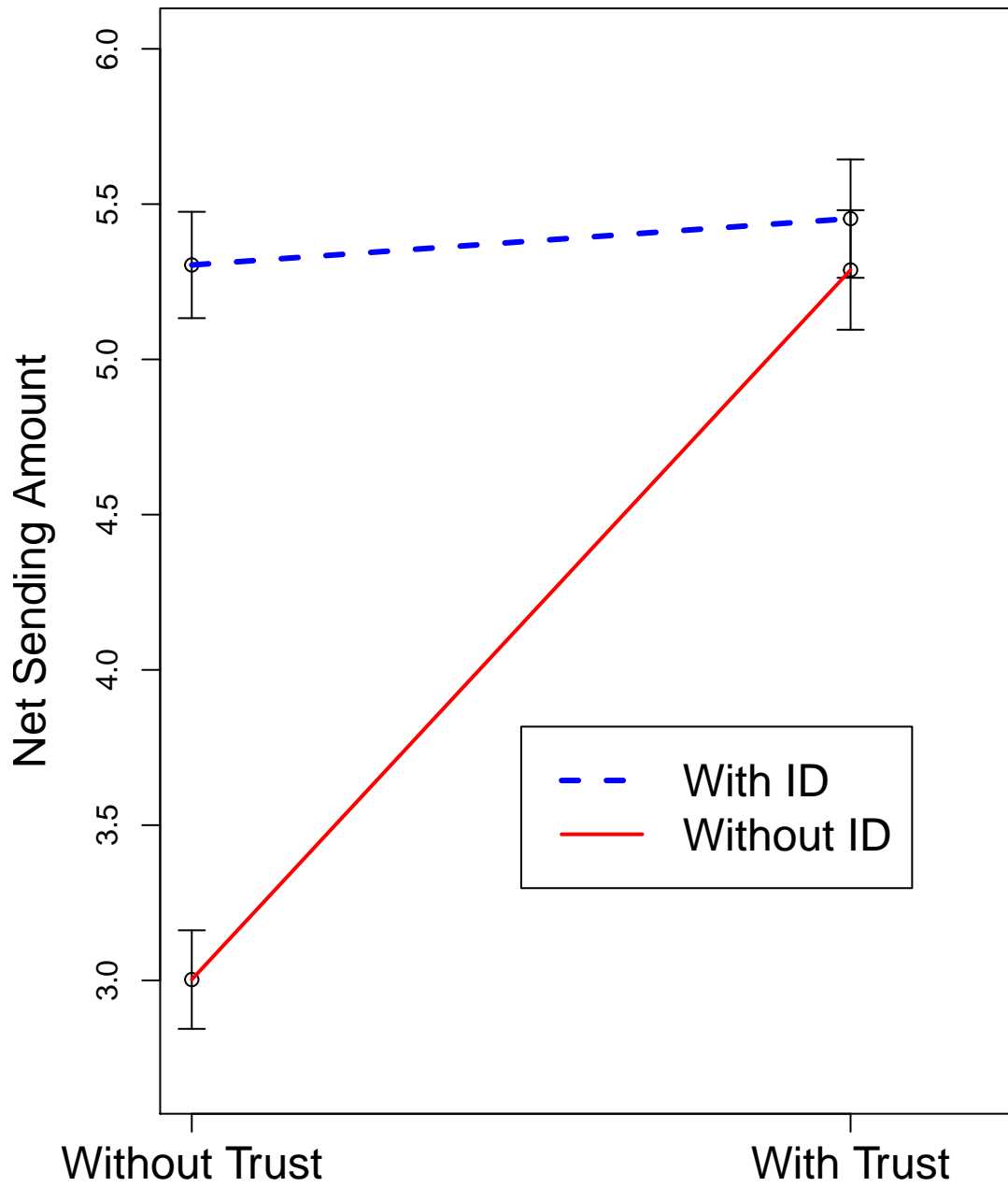
```
plotGameMetrics(simple_games$CurrGameProfit, id_games$CurrGameProfit,  
  score_games$CurrGameProfit, combine_games$CurrGameProfit,  
  metric_name = "Profit per games")
```

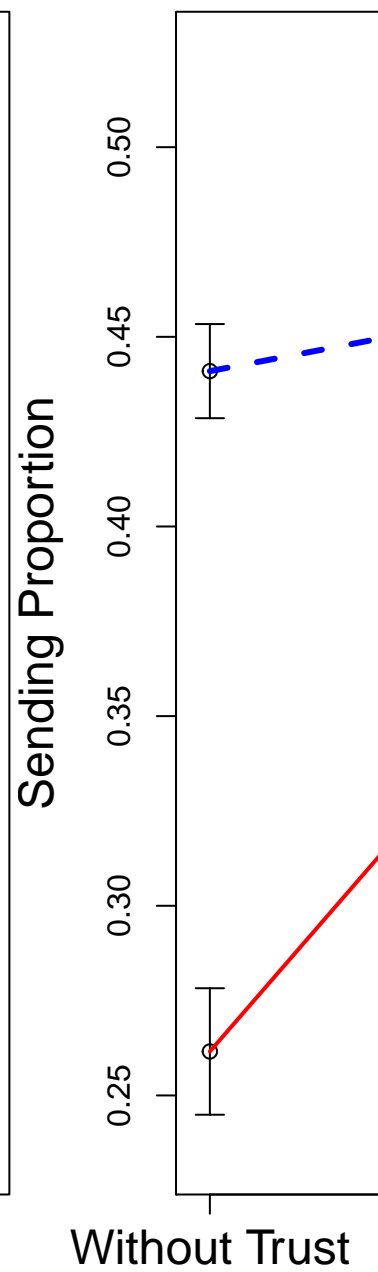
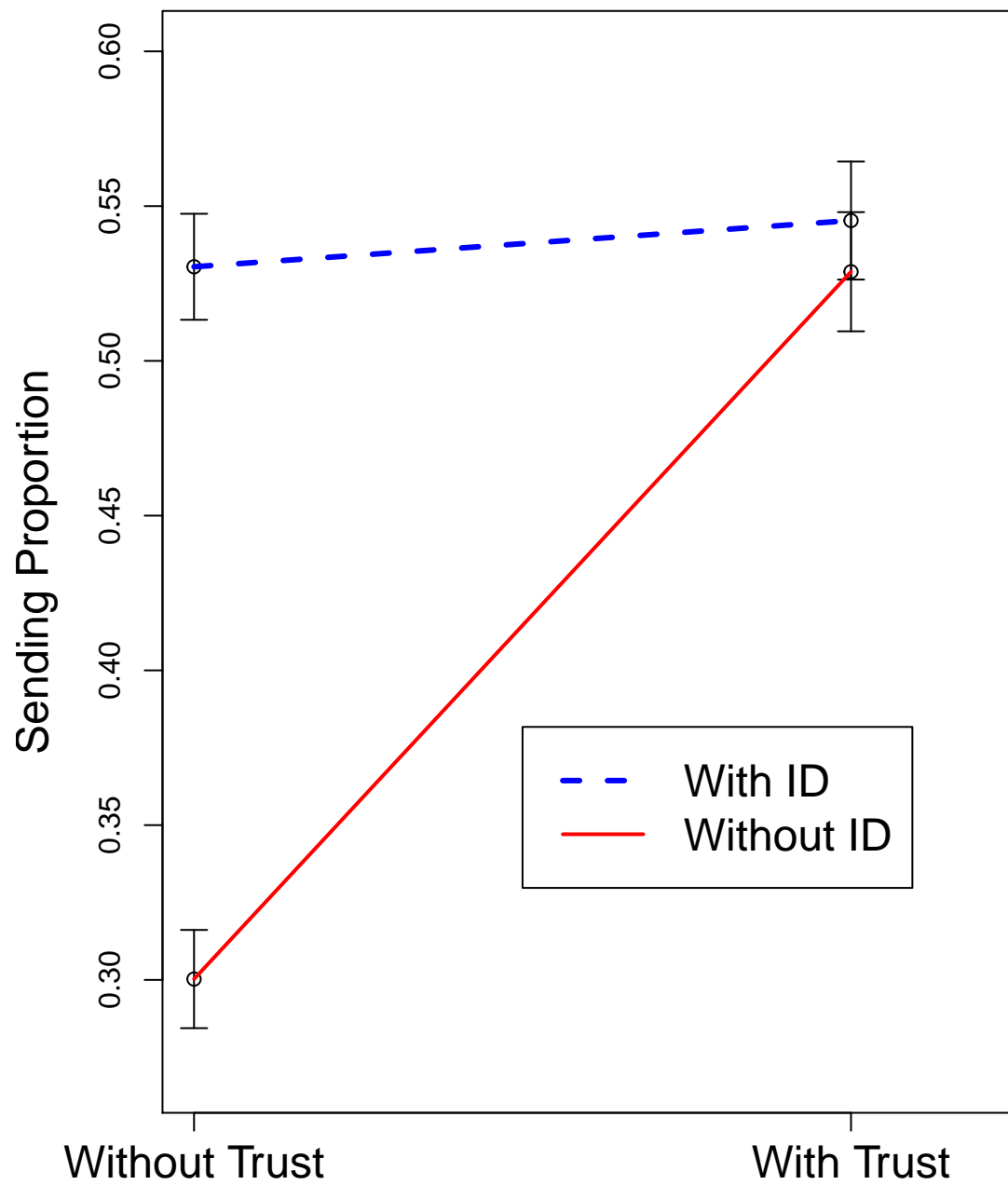




Date: Tue 7-Jun-2017

We divide the measure for sender and receiver and plot them separately





# Comparing data

## Giangiaco

We also collected data from another repeated simple games done by “Bravo, Giangiacomo, Flaminio Squazzoni, and Riccardo Boero.” Trust and partner selection in social networks: An experimentally grounded model.” Social Networks 34, no. 4 (2012): 481-492.” and we want to compare them.

```
require(BSDA)
data2 <- read.csv("./all_data/Data2.csv")

# Convert ID to universal ID
data2$id <- data2$id + (as.numeric(data2$treatment) -
  1) * 36

means <- as.numeric()
std_errors <- as.numeric()

# print ('Comparing two dataset for SENDERS') print
# (z.test(data2[!is.na(data2$daAaB),]$daAaB,
# sigma.x = 6.8)) print
# (z.test(simple_games[simple_games$Type ==
# 0,]$Contribution, sigma.x = 6.8))

# For our simple game senders
sent1 = aggregate(simple_games[simple_games$Type ==
  0,]$Contribution, list(simple_games[simple_games$Type ==
  0,]$Subject, simple_games[simple_games$Type ==
  0,]$Date), mean)
means = c(means, mean(sent1$x))
std_errors = c(std_errors, pop.sd(sent1$x)/sqrt(length(sent1$x)))

# For Giangiacomo senders
sent2 = aggregate(data2[!is.na(data2$daAaB),]$daAaB,
  list(data2[!is.na(data2$daAaB),]$id), mean)
means = c(means, mean(sent2$x))
std_errors = c(std_errors, pop.sd(sent2$x)/sqrt(length(sent2$x)))

# Compare data
print(t.test(sent1$x, sent2$x))

##
## Welch Two Sample t-test
##
## data: sent1$x and sent2$x
## t = -0.99144, df = 45.3, p-value = 0.3267
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.3726571 0.4669504
## sample estimates:
## mean of x mean of y
## 3.032332 3.485185

# print ('Comparing two dataset for RECEIVERS')
# print (z.test(data2[!is.na(data2$daBaA),]$daBaA,
```

```

# sigma.x = 6.8)) print
# (z.test(simple_games[simple_games$Type ==
# 1,]$Contribution, sigma.x = 6.8))

# For our simple game receivers
sent1 = aggregate(simple_games[simple_games$Type ==
  1,]$Contribution, list(simple_games[simple_games$Type ==
  1,]$Subject, simple_games[simple_games$Type ==
  1,]$Date), mean)
means = c(means, mean(sent1$x))
std_errors = c(std_errors, pop.sd(sent1$x)/sqrt(length(sent1$x)))

# For Giangiacomo receivers
sent2 = aggregate(data2[!is.na(data2$daBaA), ]$daBaA,
  list(data2[!is.na(data2$daBaA), ]$id), mean)
means = c(means, mean(sent2$x))
std_errors = c(std_errors, pop.sd(sent2$x)/sqrt(length(sent2$x)))

print(t.test(sent1$x, sent2$x))

##
## Welch Two Sample t-test
##
## data: sent1$x and sent2$x
## t = -0.88156, df = 50.526, p-value = 0.3822
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.3095888 0.5105319
## sample estimates:
## mean of x mean of y
## 2.393064 2.792593

# Plotting comparison
xs = c(0.75, 1.25, 2.75, 3.25)
# plot (x = xs, y = means, ylab = 'Average sending
# amount', main = 'Comparing two datasets
# w/standard errors', xaxt = 'n', xlim = c(0,8),
# ylim = c(0,4), xlab = '')
plot(x = xs, y = means, ylab = "Average sending amount",
  main = "Comparing Simple Game data", xaxt = "n",
  xlim = c(0, 8), ylim = c(0, 4), xlab = "")
axis(1, at = 1:5, labels = c("Sender", "", "Receiver",
  "", ""), las = 2)
segments(x0 = 0.75, y0 = means[1] - std_errors[1],
  x1 = 0.75, y1 = means[1] + std_errors[1], col = "red")
segments(x0 = 0.7, y0 = means[1] - std_errors[1], x1 = 0.8,
  y1 = means[1] - std_errors[1], col = "red")
segments(x0 = 0.7, y0 = means[1] + std_errors[1], x1 = 0.8,
  y1 = means[1] + std_errors[1], col = "red")

segments(x0 = 1.25, y0 = means[2] - std_errors[2],
  x1 = 1.25, y1 = means[2] + std_errors[2], col = "blue",
  lty = 2)
segments(x0 = 1.2, y0 = means[2] - std_errors[2], x1 = 1.3,

```

```

    y1 = means[2] - std_errors[2], col = "blue")
segments(x0 = 1.2, y0 = means[2] + std_errors[2], x1 = 1.3,
    y1 = means[2] + std_errors[2], col = "blue")

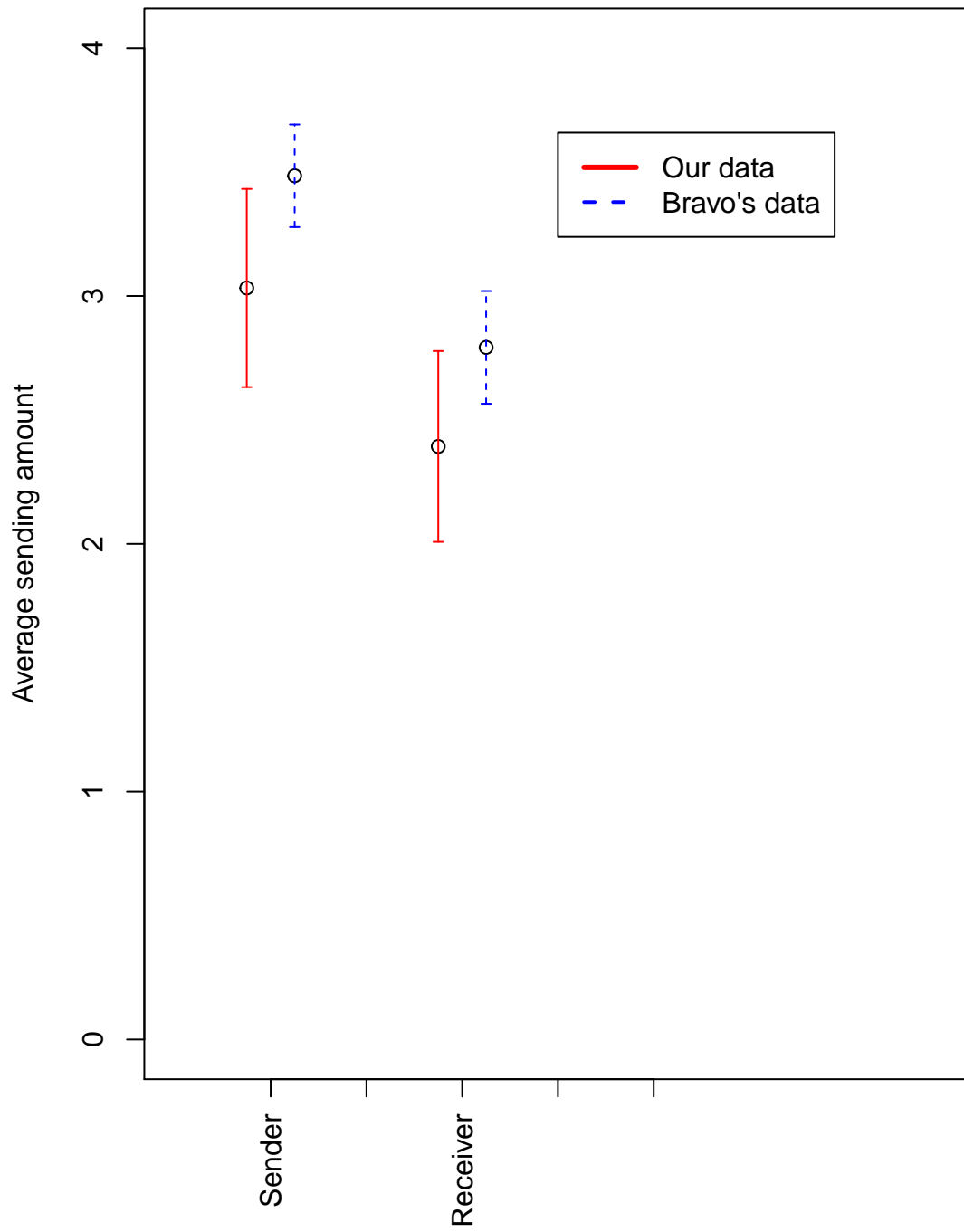
segments(x0 = 2.75, y0 = means[3] - std_errors[3],
    x1 = 2.75, y1 = means[3] + std_errors[3], col = "red")
segments(x0 = 2.7, y0 = means[3] - std_errors[3], x1 = 2.8,
    y1 = means[3] - std_errors[3], col = "red")
segments(x0 = 2.7, y0 = means[3] + std_errors[3], x1 = 2.8,
    y1 = means[3] + std_errors[3], col = "red")

segments(x0 = 3.25, y0 = means[4] - std_errors[4],
    x1 = 3.25, y1 = means[4] + std_errors[4], col = "blue",
    lty = 2)
segments(x0 = 3.2, y0 = means[4] - std_errors[4], x1 = 3.3,
    y1 = means[4] - std_errors[4], col = "blue")
segments(x0 = 3.2, y0 = means[4] + std_errors[4], x1 = 3.3,
    y1 = means[4] + std_errors[4], col = "blue")

legend(4, max(means) * 1.05, c("Our data", "Bravo's data"),
    lty = c(1, 2), lwd = c(3, 2), col = c("red", "blue"))

```

## Comparing Simple Game data



## Dubois

Analyze group effect of Dubois's data

```
dubois <- read.csv("./all_data/data_dubois.csv", sep = ";")
dubois$group <- dubois$group + dubois$treatment * 6
dubois$player_uid <- dubois$player_uid + dubois$treatment *
  36
dubois$group <- as.factor(dubois$group)
dubois$treatment <- as.factor(dubois$treatment)
dubois$player_uid <- as.factor(dubois$player_uid)

# Anova analysis per each interaction res1 <- lm
# (sent ~ treatment + treatment/group, data =
# dubois) print ('Per interaction') print
# (anova(res1))

# Anova analysis per each user
user_sent <- as.vector(aggregate(dubois$sent, list(dubois$player_uid),
  mean)[, "x"])
user_sent_back <- as.vector(aggregate(dubois$sent_back,
  list(dubois$player_uid), mean)[, "x"])
user_reciprocity <- as.vector(aggregate(dubois$sent_back/dubois$received,
  list(dubois$player_uid), mean, na.rm = TRUE)[,
  "x"])
user_sender_payoff <- as.vector(aggregate(dubois$returned -
  dubois$sent, list(dubois$player_uid), mean, na.rm = TRUE)[,
  "x"])
user_receiver_payoff <- as.vector(aggregate(dubois$received -
  dubois$sent_back, list(dubois$player_uid), mean,
  na.rm = TRUE)[, "x"])
treatment <- as.factor(c(rep(0, 36), rep(1, 36), rep(2,
  36)))
group <- as.factor(rep(1:18, each = 6))

res2 <- lm(user_sent ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
  df2 = a$Df[2])
print(a)
```

## Analysis of Variance Table

##

## Response: user\_sent

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment    2  27.941  13.9707  2.0951 0.15761
## treatment:group 15 100.023   6.6682  2.0706 0.01846 *
## Residuals    90 289.836   3.2204
```

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
res2 <- lm(user_reciprocity ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
```



```

a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
  df2 = a$Df[2])
print(a)

## Analysis of Variance Table
##
## Response: user_reciprocity
##              Df Sum Sq Mean Sq F value Pr(>F)
## treatment      2  0.06941  0.034704   1.7882 0.2011
## treatment:group 15  0.29112  0.019408   0.7685 0.7083
## Residuals      90  2.27300  0.025256

res2 <- lm(user_sender_payoff ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
  df2 = a$Df[2])
print(a)

## Analysis of Variance Table
##
## Response: user_sender_payoff
##              Df Sum Sq Mean Sq F value    Pr(>F)
## treatment      2  21.006  10.5030   2.4575   0.1194
## treatment:group 15  64.108   4.2739   3.7518 4.119e-05 ***
## Residuals      90 102.525   1.1392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

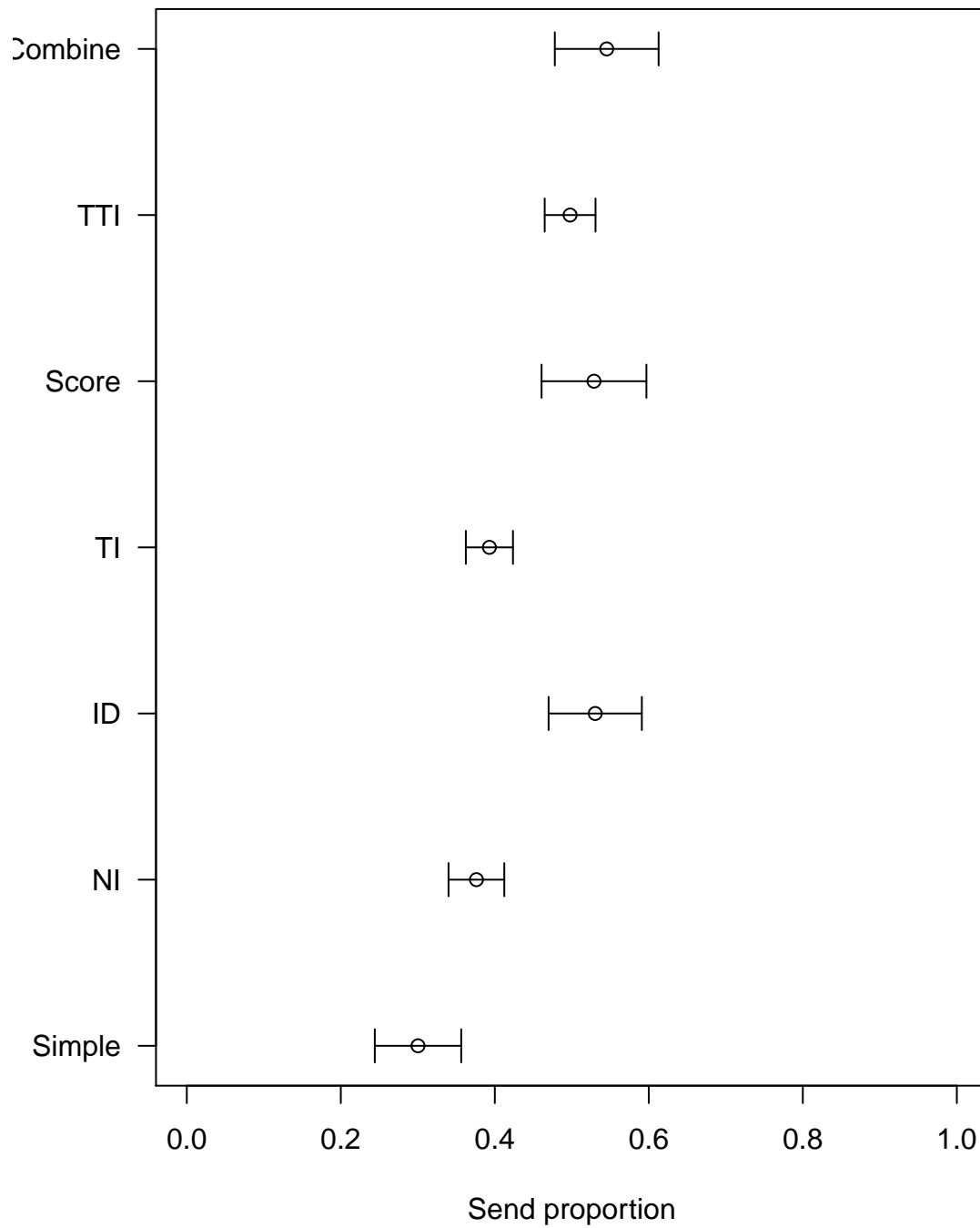
res2 <- lm(user_receiver_payoff ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
  df2 = a$Df[2])
print(a)

## Analysis of Variance Table
##
## Response: user_receiver_payoff
##              Df Sum Sq Mean Sq F value    Pr(>F)
## treatment      2  58.57 29.2853   1.0625   0.3702
## treatment:group 15 413.44 27.5626   5.5594 7.684e-08 ***
## Residuals      90 446.20  4.9578
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

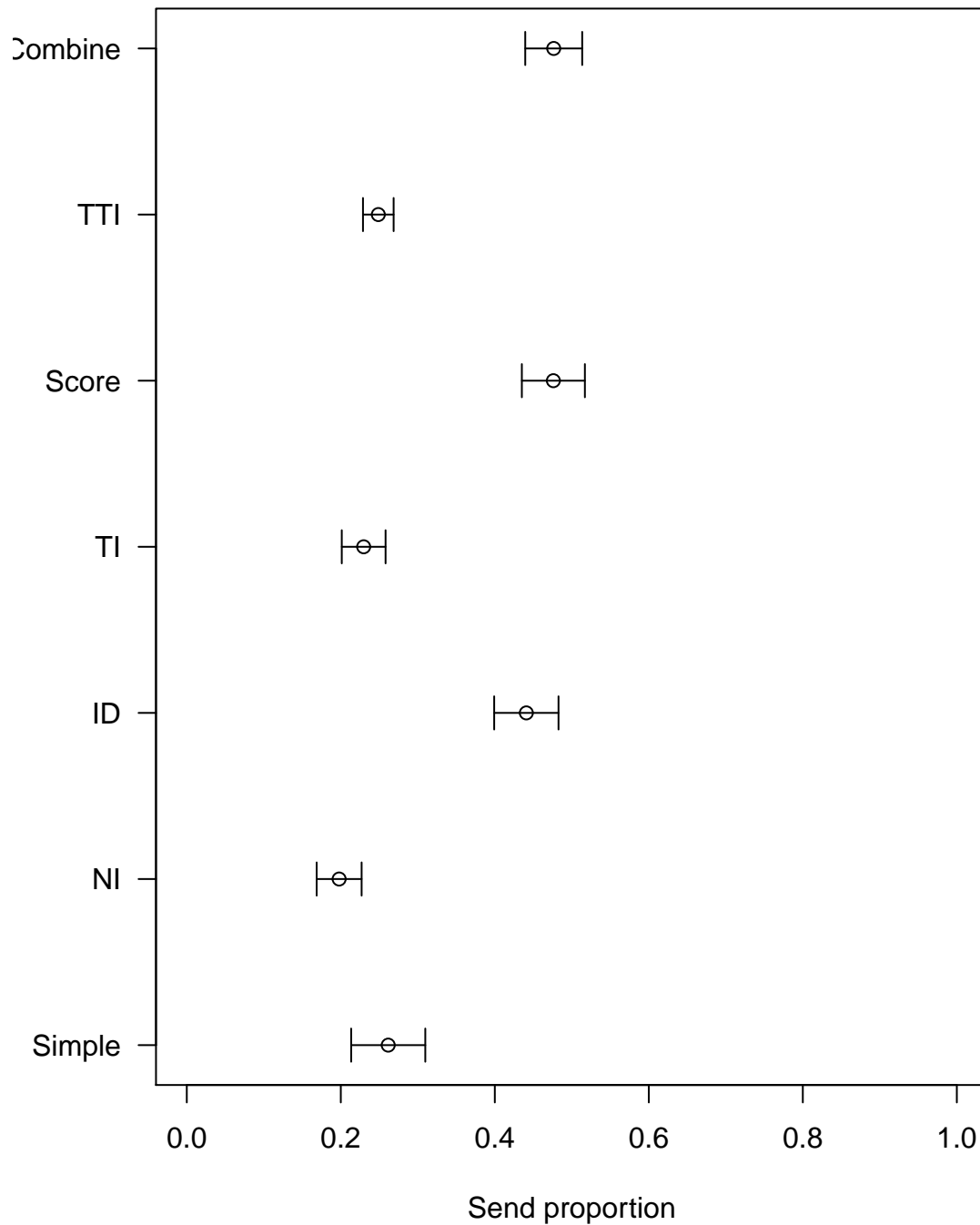
Comparing Dubois data

### Compare with Dubois of type SENDER



##  
##  
## \pagebreak

## Compare with Dubois of type RECEIVER



```
##  
##  
## \pagebreak  
means <- as.numeric()  
std_errors <- as.numeric()  
for (type in 0:1) {  
  data_dubois <- as.numeric()  
  if (type == 0) {
```

```

    data_dubois <- user_sent/10
  } else if (type == 1) {
    data_dubois <- user_reciprocity
  }
  means <- c(means, mean(simple_games[simple_games$Type ==
    type & simple_games$my_send_proportional >=
    0, ]$my_send_proportional))
  std_errors <- c(std_errors, pop.sd(simple_games[simple_games$Type ==
    type & simple_games$my_send_proportional >=
    0, ]$my_send_proportional)/sqrt(30))

  means <- c(means, mean(data_dubois[1:36]))
  std_errors <- c(std_errors, pop.sd(data_dubois[1:36])/sqrt(36))
}

xs = c(0.75, 1.25, 2.75, 3.25)
# plot (x = xs, y = means, ylab = 'Average sending
# amount', main = 'Comparing two datasets
# w/standard errors', xaxt = 'n', xlim = c(0,8),
# ylim = c(0,4), xlab = '')
plot(x = xs, y = means, ylab = "Average sending proportion",
  main = "Comparing Simple Game data", xaxt = "n",
  xlim = c(0, 8), ylim = c(0, 0.5), xlab = "")
axis(1, at = 1:5, labels = c("Sender", "", "Receiver",
  "", ""), las = 2)
segments(x0 = 0.75, y0 = means[1] - std_errors[1],
  x1 = 0.75, y1 = means[1] + std_errors[1], col = "red")
segments(x0 = 0.7, y0 = means[1] - std_errors[1], x1 = 0.8,
  y1 = means[1] - std_errors[1], col = "red")
segments(x0 = 0.7, y0 = means[1] + std_errors[1], x1 = 0.8,
  y1 = means[1] + std_errors[1], col = "red")

segments(x0 = 1.25, y0 = means[2] - std_errors[2],
  x1 = 1.25, y1 = means[2] + std_errors[2], col = "blue",
  lty = 2)
segments(x0 = 1.2, y0 = means[2] - std_errors[2], x1 = 1.3,
  y1 = means[2] - std_errors[2], col = "blue")
segments(x0 = 1.2, y0 = means[2] + std_errors[2], x1 = 1.3,
  y1 = means[2] + std_errors[2], col = "blue")

segments(x0 = 2.75, y0 = means[3] - std_errors[3],
  x1 = 2.75, y1 = means[3] + std_errors[3], col = "red")
segments(x0 = 2.7, y0 = means[3] - std_errors[3], x1 = 2.8,
  y1 = means[3] - std_errors[3], col = "red")
segments(x0 = 2.7, y0 = means[3] + std_errors[3], x1 = 2.8,
  y1 = means[3] + std_errors[3], col = "red")

segments(x0 = 3.25, y0 = means[4] - std_errors[4],
  x1 = 3.25, y1 = means[4] + std_errors[4], col = "blue",
  lty = 2)
segments(x0 = 3.2, y0 = means[4] - std_errors[4], x1 = 3.3,
  y1 = means[4] - std_errors[4], col = "blue")
segments(x0 = 3.2, y0 = means[4] + std_errors[4], x1 = 3.3,

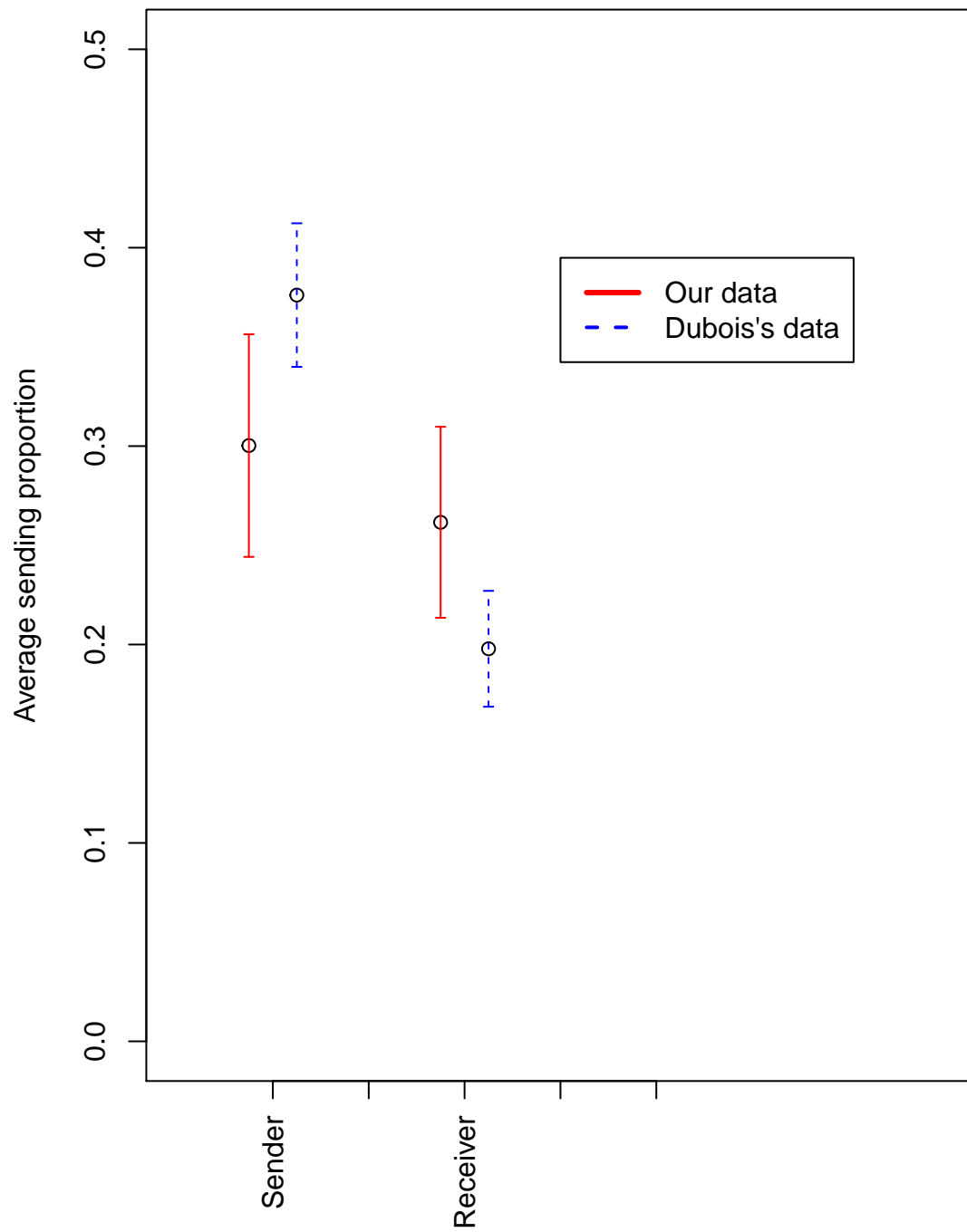
```

```

y1 = means[4] + std_errors[4], col = "blue")
legend(4, max(means) * 1.05, c("Our data", "Dubois's data"),
      lty = c(1, 2), lwd = c(3, 2), col = c("red", "blue"))

```

## Comparing Simple Game data



## Plot net sending amount of three datasets in one figure

```
## [1] "Test sending amount of senders"

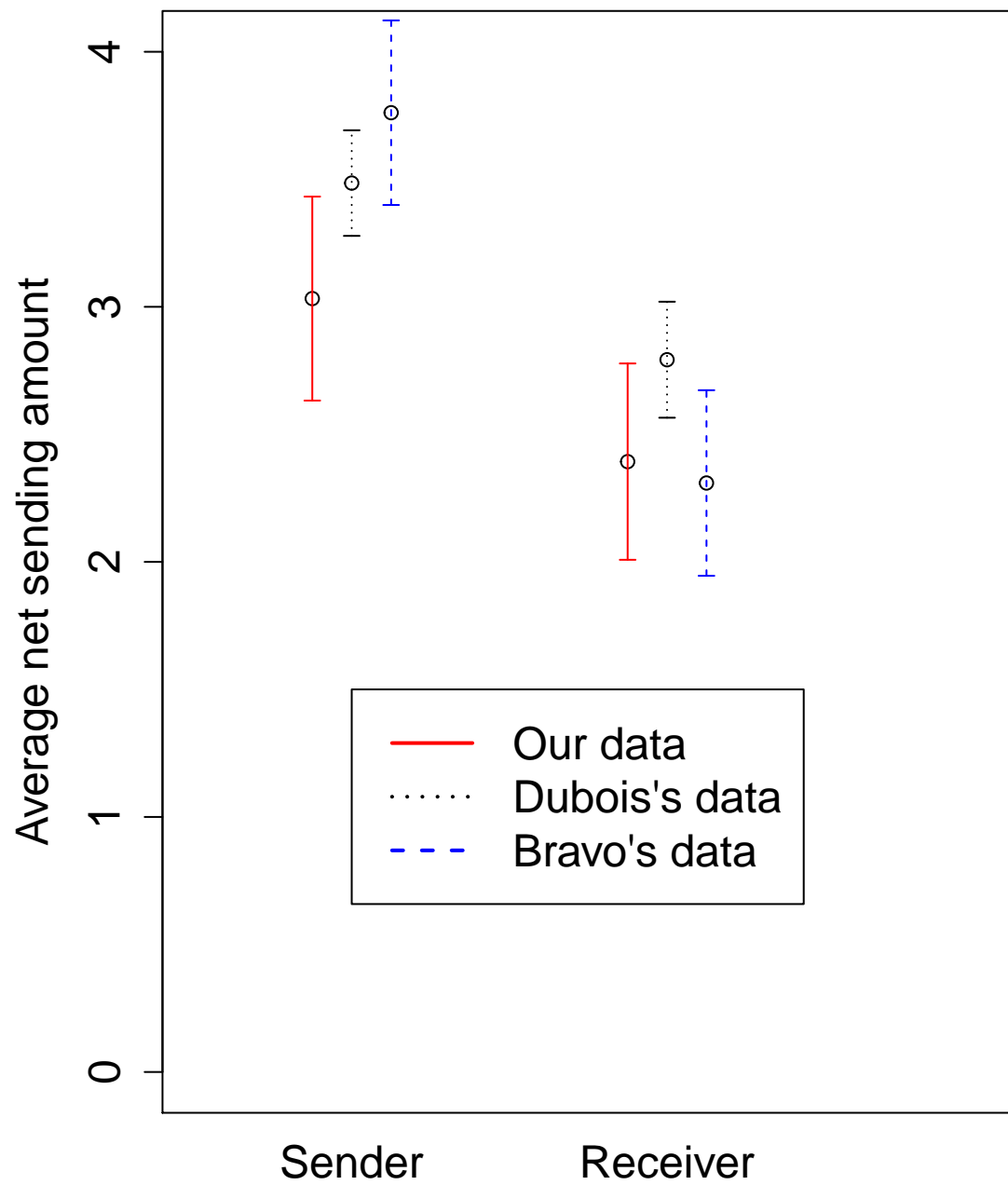
##
## Welch Two Sample t-test
##
## data: sent1$x and sent2$x
## t = -0.99144, df = 45.3, p-value = 0.3267
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.3726571 0.4669504
## sample estimates:
## mean of x mean of y
## 3.032332 3.485185

##
## Welch Two Sample t-test
##
## data: sent1$x and sent3
## t = -1.3304, df = 61.606, p-value = 0.1883
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.823901 0.366342
## sample estimates:
## mean of x mean of y
## 3.032332 3.761111

## [1] "Compare sending amount of receivers"

##
## Welch Two Sample t-test
##
## data: sent1$x and sent2$x
## t = -0.88156, df = 50.526, p-value = 0.3822
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.3095888 0.5105319
## sample estimates:
## mean of x mean of y
## 2.393064 2.792593

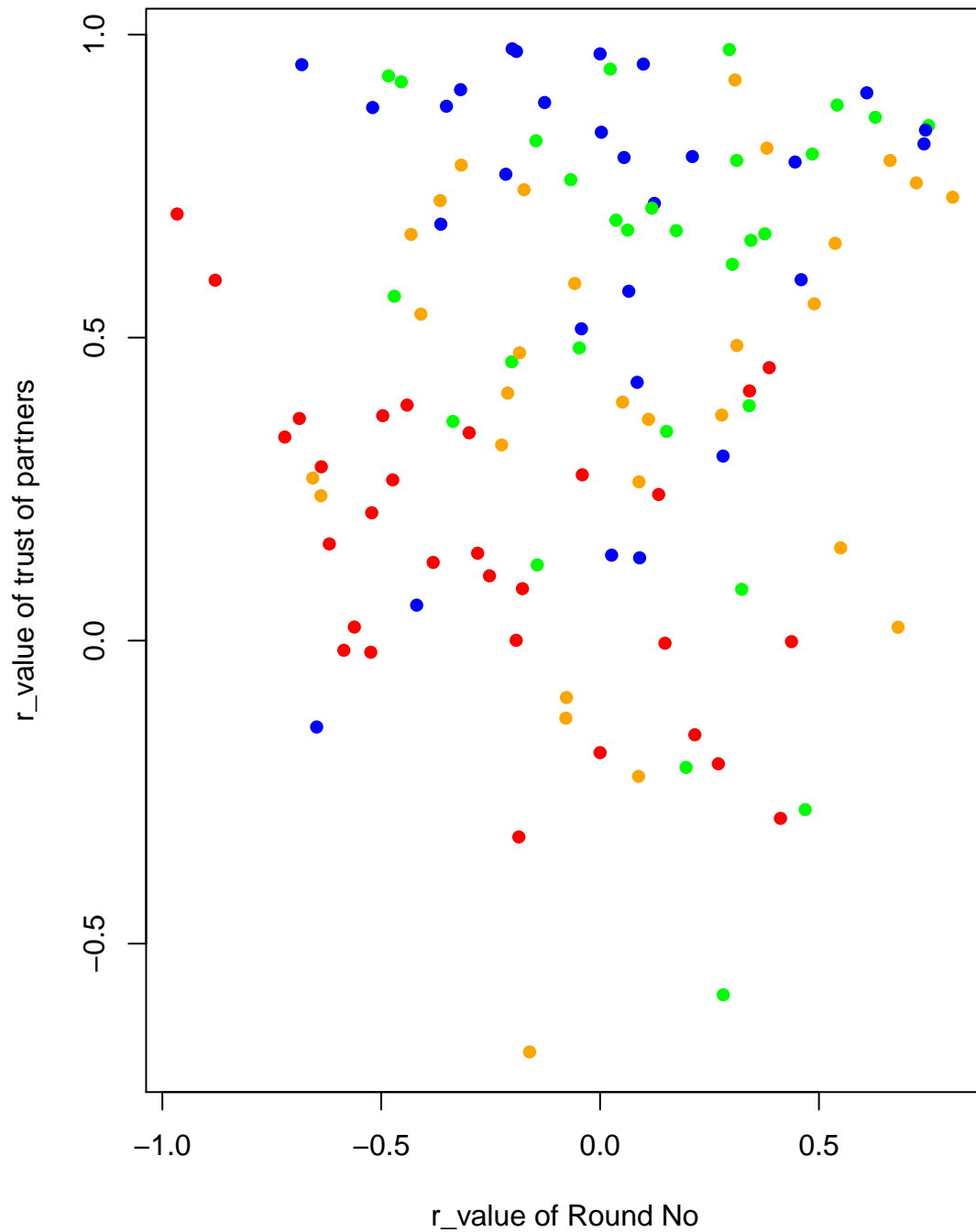
##
## Welch Two Sample t-test
##
## data: sent1$x and sent3
## t = 0.15581, df = 62.523, p-value = 0.8767
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.9911877 1.1587975
## sample estimates:
## mean of x mean of y
## 2.393064 2.309259
```



## Behavior over time

After reading the questionnaire, there is a hypothesis that, people send less in the end of game (i.e, people learn the length of the game and adapt to the game). However, after running the regression test between sending behavior and period, there is no evidence to prove the hypothesis.

### r\_value of behavior over time of SENDER



```
##           NA      NA      NA
## 1  Slope Without trust 0.02183 0.00639
## 2  Slope   With trust 0.02256 0.0124
```



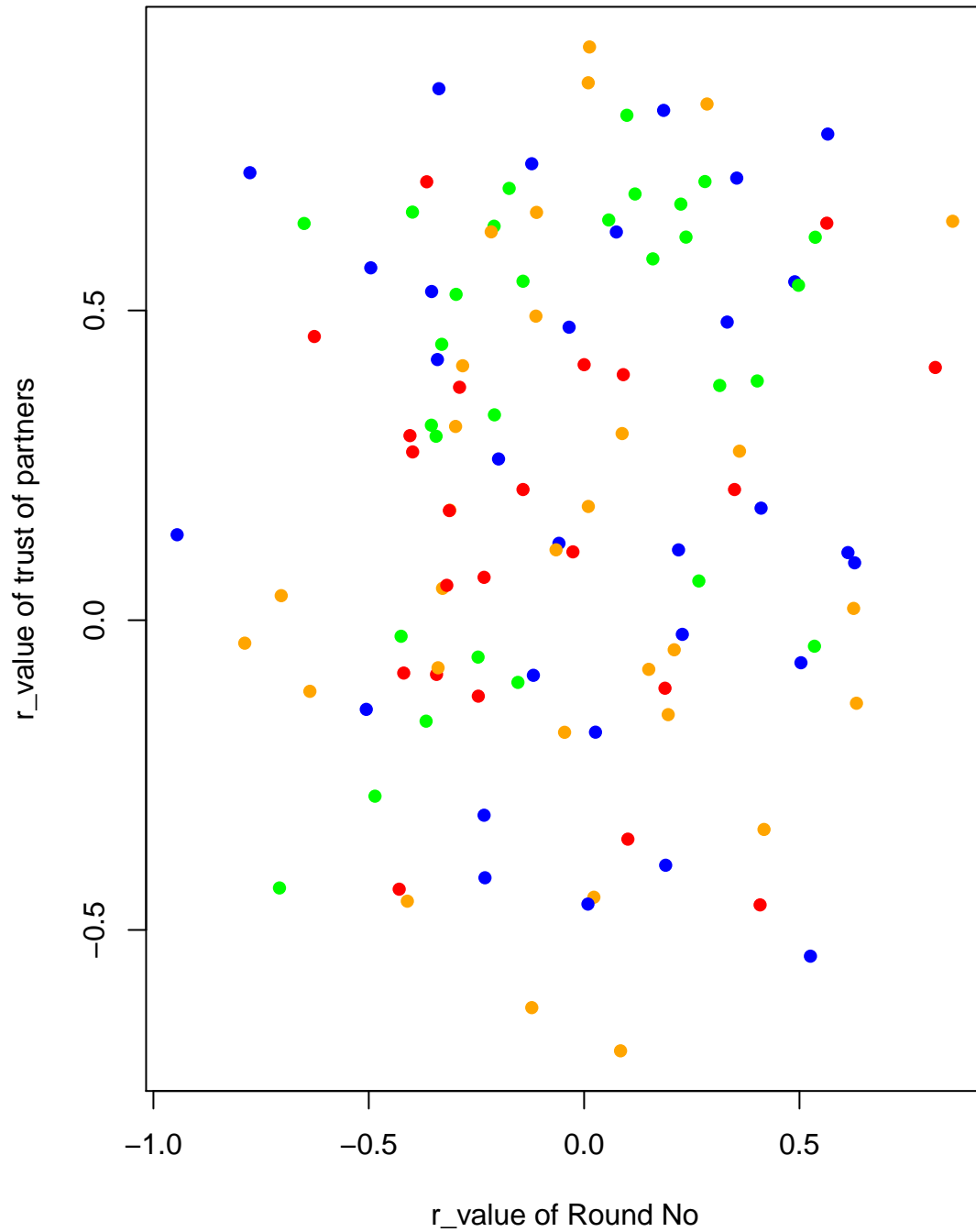
```

## 3 Intercept Without trust 0.50555 0.3186
## 4 Intercept With trust 0.31454 0.20829
## 5 r_value Without trust 0.63535 0.10645
## 6 r_value With trust 0.61977 0.1123
## 7 r_value Without ID 0.63808 0.1228
## 8 r_value With ID 0.62 0.11
## [1] "Anova analysis for slope"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
## SS num Df Error SS den Df F
## (Intercept) 0.00000193 1 0.0034749 24 0.0133
## GroupID 0.00086383 4 0.0034749 24 1.4916
## SHOW_TRUST 0.00049717 1 0.0034653 24 4.1313
## GroupID:SHOW_TRUST 0.00048138 4 0.0034653 24 0.8335
## SHOW_ID 0.00058784 1 0.0027101 24 2.9728
## GroupID:SHOW_ID 0.00079096 4 0.0027101 24 1.7511
## SHOW_TRUST:SHOW_ID 0.00099708 1 0.0028265 24 4.8548
## GroupID:SHOW_TRUST:SHOW_ID 0.00082152 4 0.0028265 24 1.7439
## Pr(>F)
## (Intercept) 0.90900
## GroupID 0.23607
## SHOW_TRUST 0.11189
## GroupID:SHOW_TRUST 0.51724
## SHOW_ID 0.15977
## GroupID:SHOW_ID 0.17177
## SHOW_TRUST:SHOW_ID 0.09231
## GroupID:SHOW_TRUST:SHOW_ID 0.17330
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "Anova analysis for intercept"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
## SS num Df Error SS den Df F
## (Intercept) 1.18242 1 1.04262 24 27.2181
## GroupID 0.29525 4 1.04262 24 1.6991
## SHOW_TRUST 0.19793 1 1.41746 24 3.9575
## GroupID:SHOW_TRUST 0.20005 4 1.41746 24 0.8468
## SHOW_ID 0.01085 1 0.83865 24 2.5412
## GroupID:SHOW_ID 0.01708 4 0.83865 24 0.1222
## SHOW_TRUST:SHOW_ID 0.03465 1 1.18402 24 0.5485
## GroupID:SHOW_TRUST:SHOW_ID 0.25264 4 1.18402 24 1.2802
## Pr(>F)
## (Intercept) 2.401e-05 ***
## GroupID 0.1831
## SHOW_TRUST 0.1175
## GroupID:SHOW_TRUST 0.5095
## SHOW_ID 0.1861
## GroupID:SHOW_ID 0.9732
## SHOW_TRUST:SHOW_ID 0.5000
## GroupID:SHOW_TRUST:SHOW_ID 0.3054
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"

```

```
## [1] "Anova analysis for R value"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
##          SS num Df Error SS den Df      F Pr(>F)
## (Intercept)      0.00368      1  1.9572    23  0.0432 0.8371
## GroupID          0.53986      4  1.9572    23  1.5860 0.2117
## SHOW_TRUST       0.31137      1  2.0440    23  3.1651 0.1498
## GroupID:SHOW_TRUST 0.39351      4  2.0440    23  1.1070 0.3771
## SHOW_ID          0.64977      1  1.7228    23 17.6350 0.0137 *
## GroupID:SHOW_ID   0.14738      4  1.7228    23  0.4919 0.7417
## SHOW_TRUST:SHOW_ID 0.48743      1  1.8761    23  2.8501 0.1666
## GroupID:SHOW_TRUST:SHOW_ID 0.68408      4  1.8761    23  2.0966 0.1141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
##
##
## \pagebreak
```

## r\_value of behavior over time of RECEIVER



```
##           NA      NA      NA
## 1  Slope Without trust 0.00826 0.00284
## 2  Slope   With trust 0.00595 0.00121
## 3 Intercept Without trust 0.72518 0.24914
## 4 Intercept   With trust 0.56743 0.15225
## 5  r_value Without trust 0.74394 0.14981
## 6  r_value   With trust 0.55454 0.04005
## 7  r_value Without ID 0.5605 0.05218
```

```

## 8   r_value      With ID    0.67    0.13
## [1] "Anova analysis for slope"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##               SS num Df   Error SS den Df      F
## (Intercept)      1.2602e-04      1 0.00116706      17 1.8357
## GroupID          3.1315e-04      4 0.00116706      17 1.1404
## SHOW_TRUST       5.9253e-05      1 0.00059534      17 0.7738
## GroupID:SHOW_TRUST 3.0629e-04      4 0.00059534      17 2.1865
## SHOW_ID          2.0170e-05      1 0.00066218      17 0.2897
## GroupID:SHOW_ID   2.7853e-04      4 0.00066218      17 1.7876
## SHOW_TRUST:SHOW_ID 1.1654e-04      1 0.00123391      17 1.8225
## GroupID:SHOW_TRUST:SHOW_ID 2.5579e-04      4 0.00123391      17 0.8810
##
##               Pr(>F)
## (Intercept)      0.1932
## GroupID          0.3710
## SHOW_TRUST       0.4287
## GroupID:SHOW_TRUST 0.1141
## SHOW_ID          0.6190
## GroupID:SHOW_ID   0.1779
## SHOW_TRUST:SHOW_ID 0.2484
## GroupID:SHOW_TRUST:SHOW_ID 0.4959
## [1] "----*-----"
## [1] "Anova analysis for intercept"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##               SS num Df Error SS den Df      F Pr(>F)
## (Intercept)      0.48013      1 0.86570      17 9.4284 0.006928
## GroupID          0.13089      4 0.86570      17 0.6426 0.639491
## SHOW_TRUST       0.01316      1 0.49867      17 0.1338 0.733030
## GroupID:SHOW_TRUST 0.39339      4 0.49867      17 3.3527 0.033854
## SHOW_ID          0.00709      1 0.65748      17 0.1179 0.748586
## GroupID:SHOW_ID   0.24064      4 0.65748      17 1.5555 0.231309
## SHOW_TRUST:SHOW_ID 0.23109      1 1.02451      17 4.1807 0.110356
## GroupID:SHOW_TRUST:SHOW_ID 0.22111      4 1.02451      17 0.9172 0.476531
##
## (Intercept)      **
## GroupID
## SHOW_TRUST
## GroupID:SHOW_TRUST      *
## SHOW_ID
## GroupID:SHOW_ID
## SHOW_TRUST:SHOW_ID
## GroupID:SHOW_TRUST:SHOW_ID
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "Anova analysis for R value"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##               SS num Df Error SS den Df      F Pr(>F)
## (Intercept)      0.004447      1 1.25021      17 0.0605 0.8087
## GroupID          0.271292      4 1.25021      17 0.9222 0.4739
## SHOW_TRUST       0.114339      1 0.55834      17 2.4233 0.1945

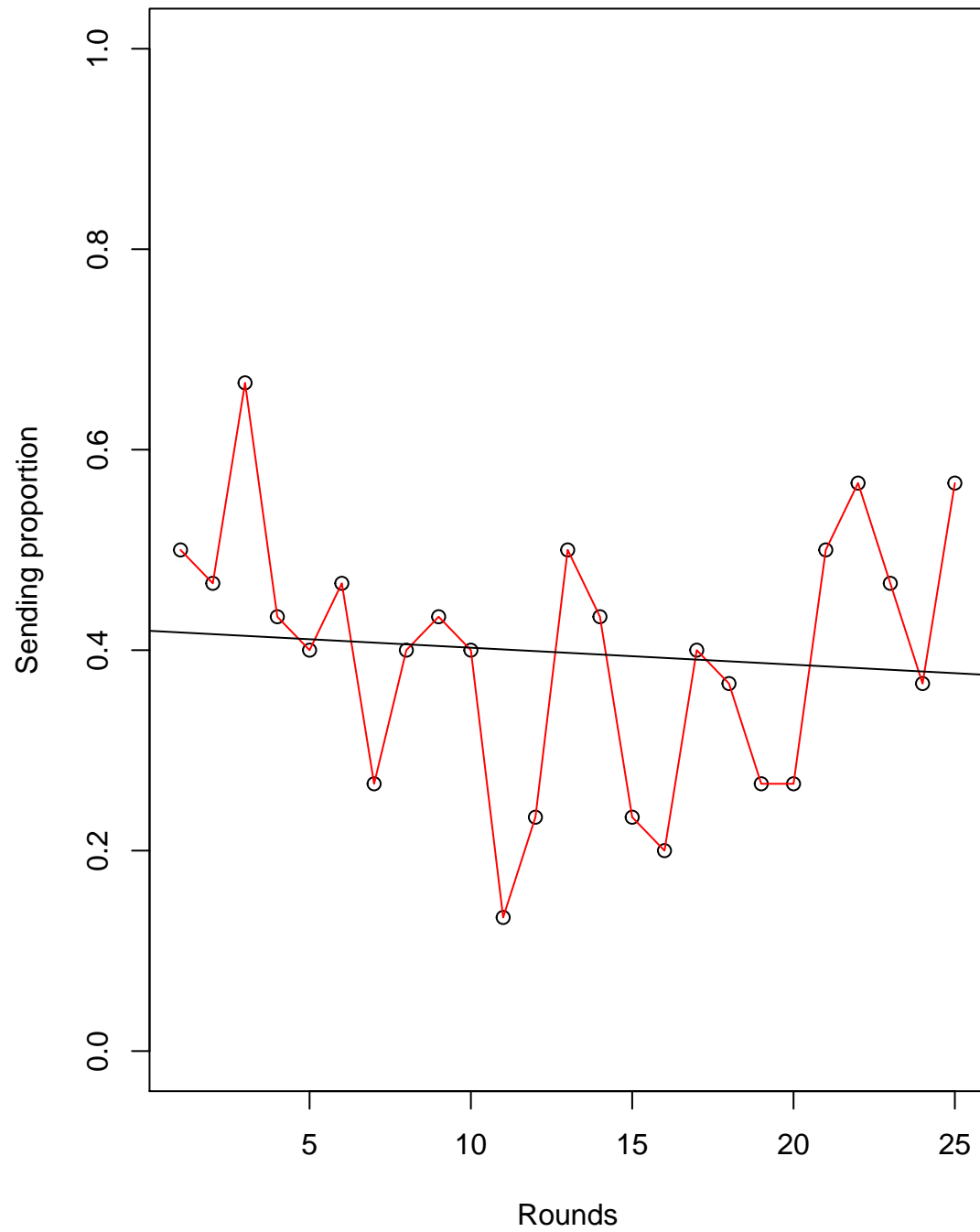
```

```

## GroupID:SHOW_TRUST      0.188735      4  0.55834      17 1.4366 0.2648
## SHOW_ID                 0.008131      1  0.72676      17 0.2571 0.6388
## GroupID:SHOW_ID        0.126522      4  0.72676      17 0.7399 0.5777
## SHOW_TRUST:SHOW_ID     0.002261      1  1.41863      17 0.0861 0.7838
## GroupID:SHOW_TRUST:SHOW_ID 0.105058      4  1.41863      17 0.3147 0.8642
## [1] "----*---"
##
##
## \pagebreak

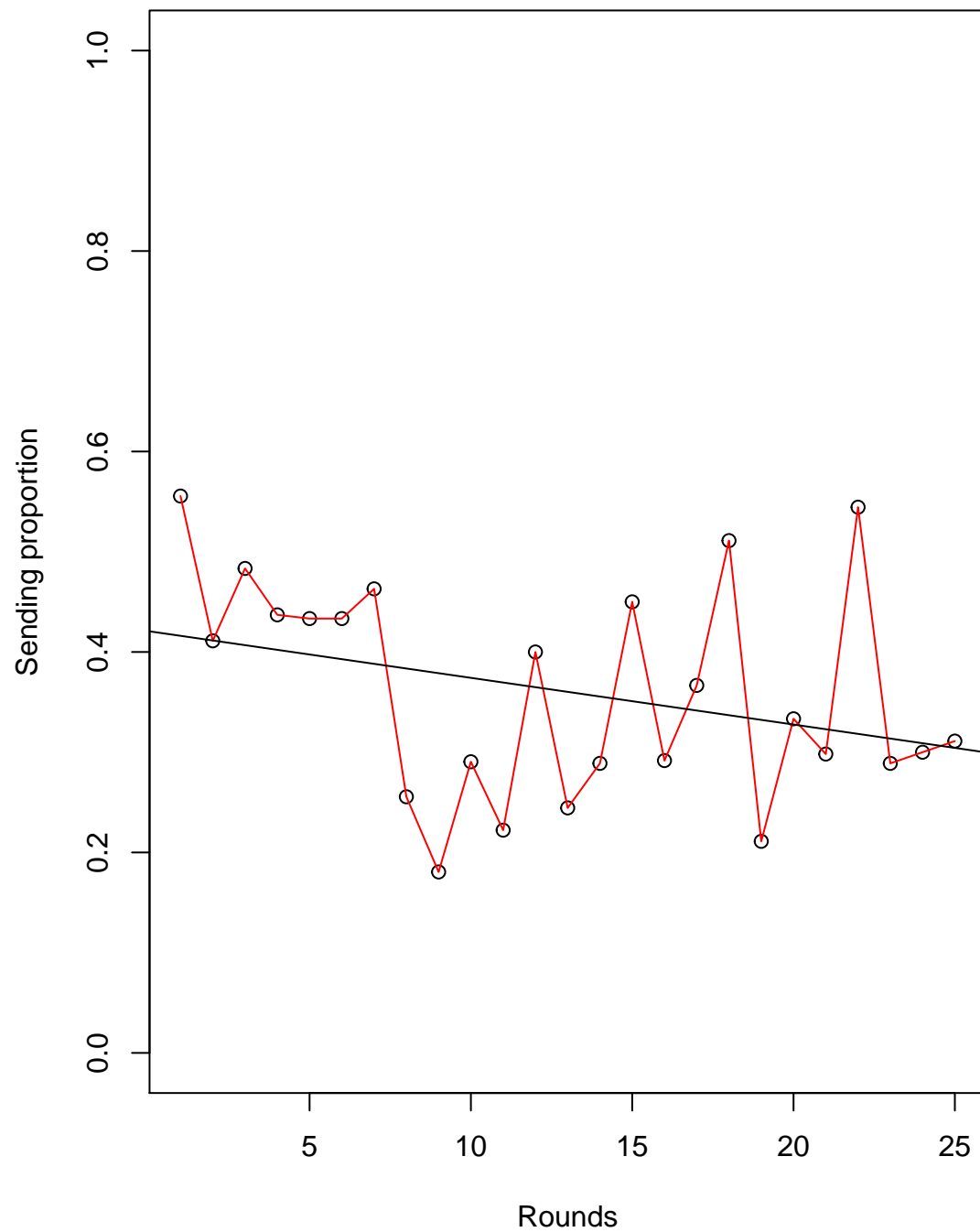
```

## Sending behavior of Simple Games over time of: SENDER



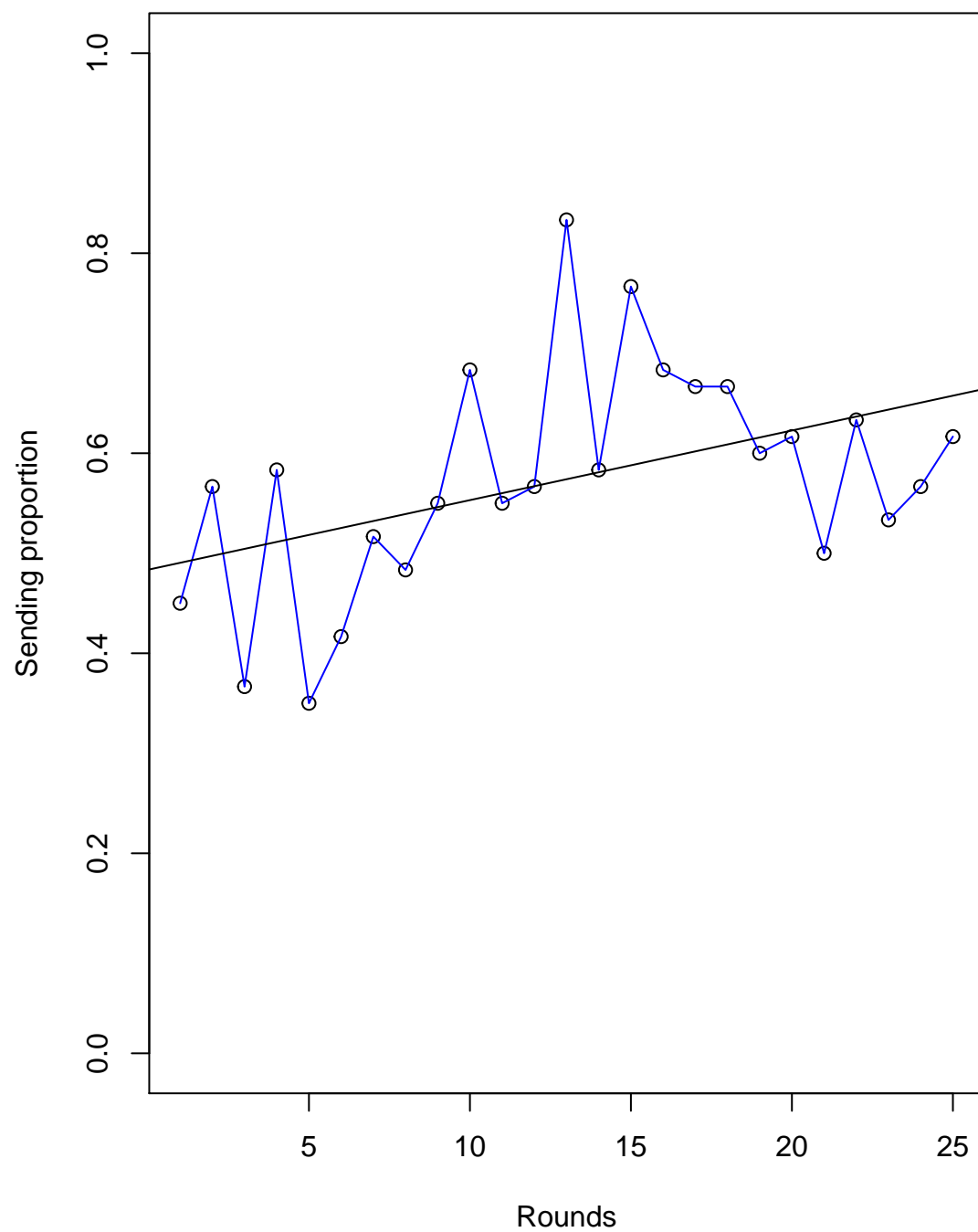
##  
##  
## \pagebreak

## Sending behavior of Simple Games over time of: RECEIVER



##  
##  
## \pagebreak

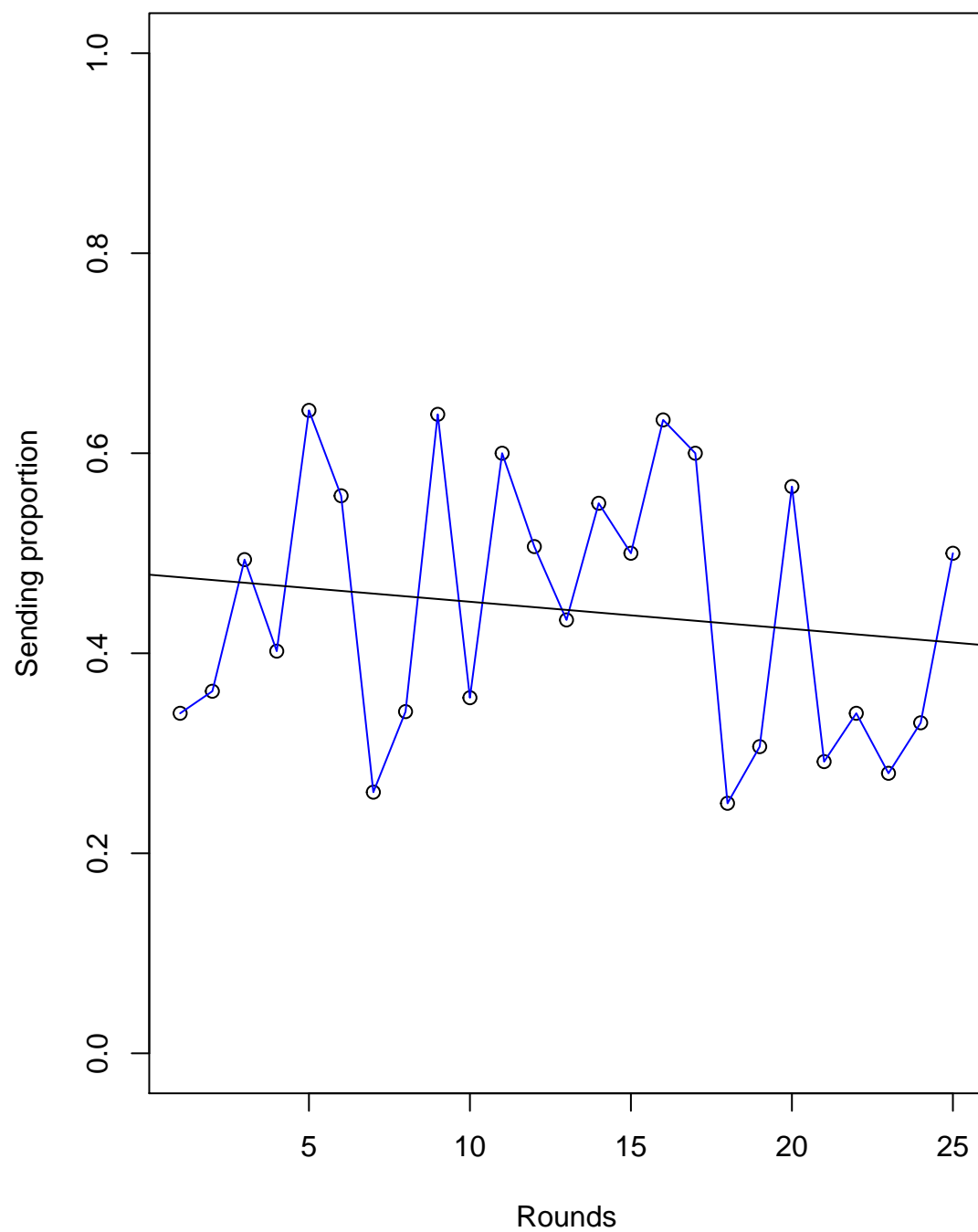
## Sending behavior of ID Games over time of: SENDER



##  
##  
## \pagebreak

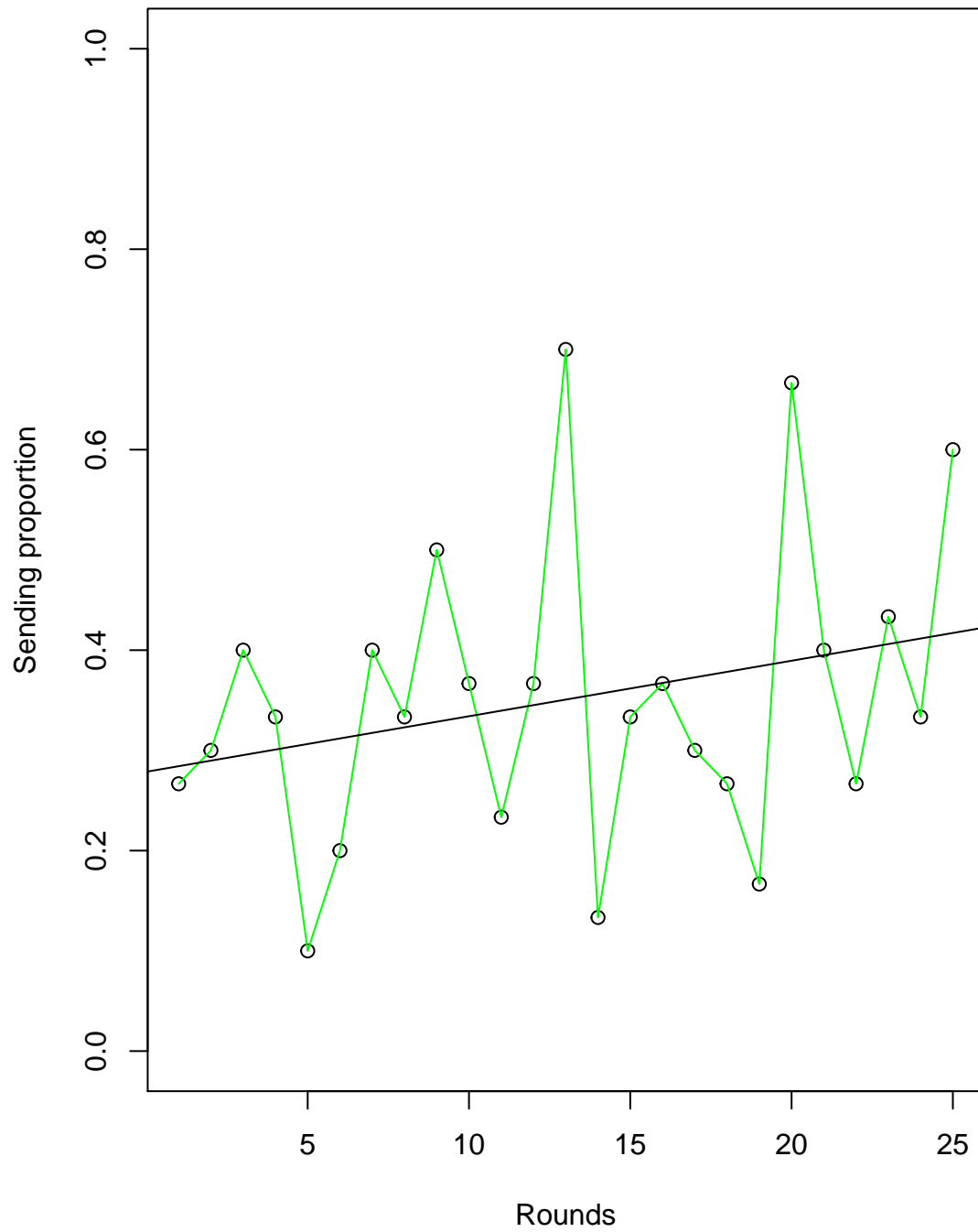


## Sending behavior of ID Games over time of: RECEIVER



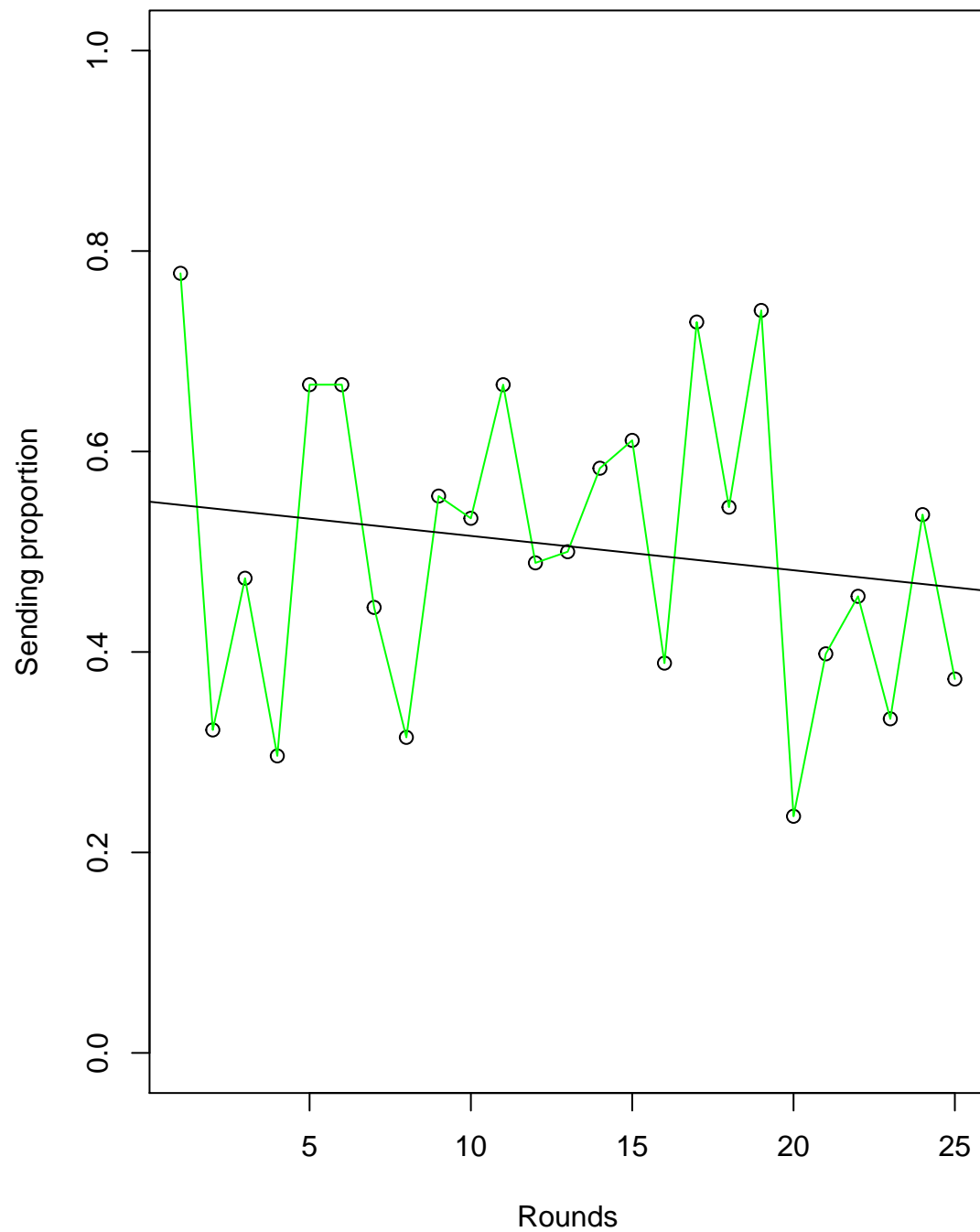
##  
##  
## \pagebreak

## Sending behavior of Score Games over time of: SENDER



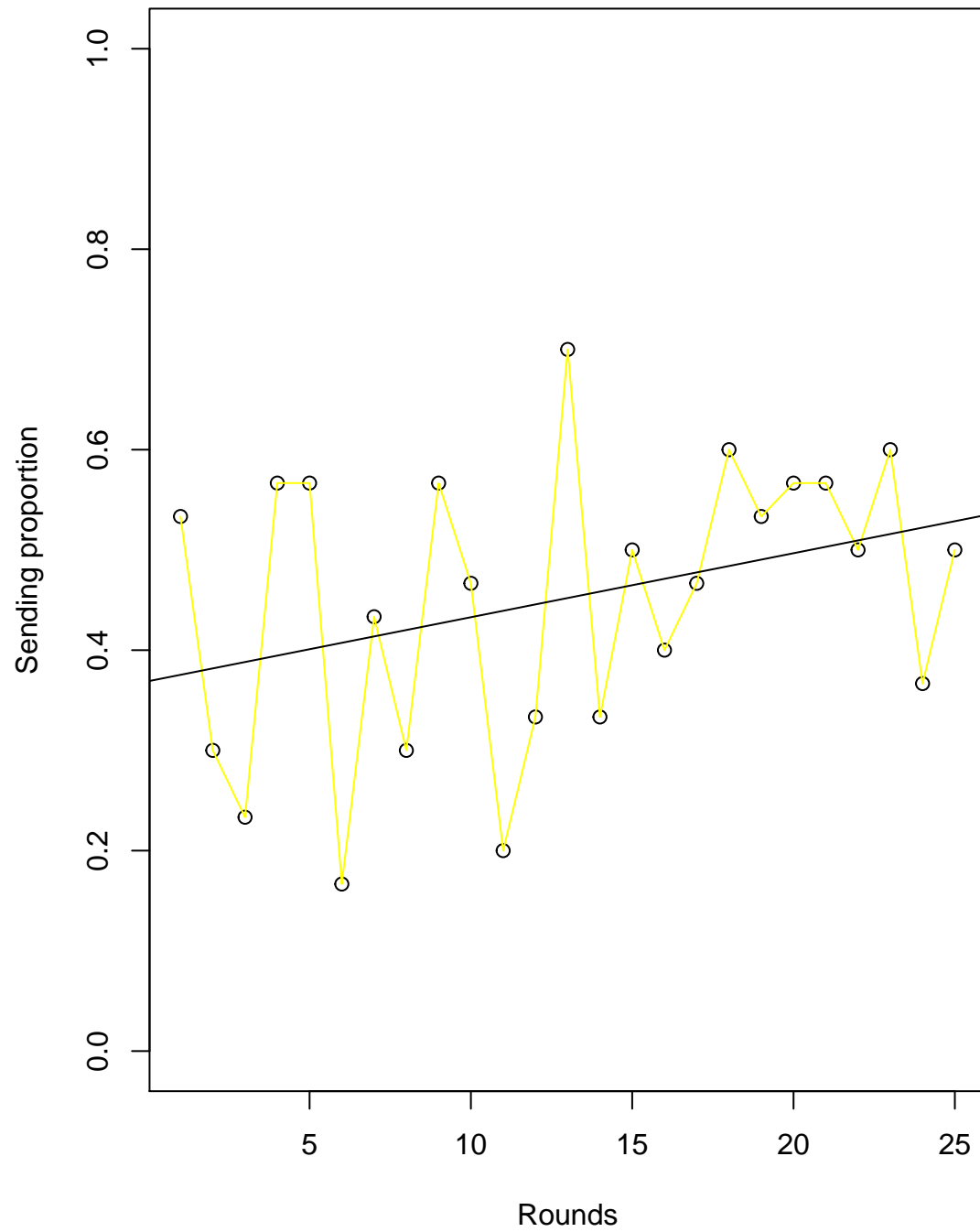
##  
##  
## \pagebreak

## Sending behavior of Score Games over time of: RECEIVER



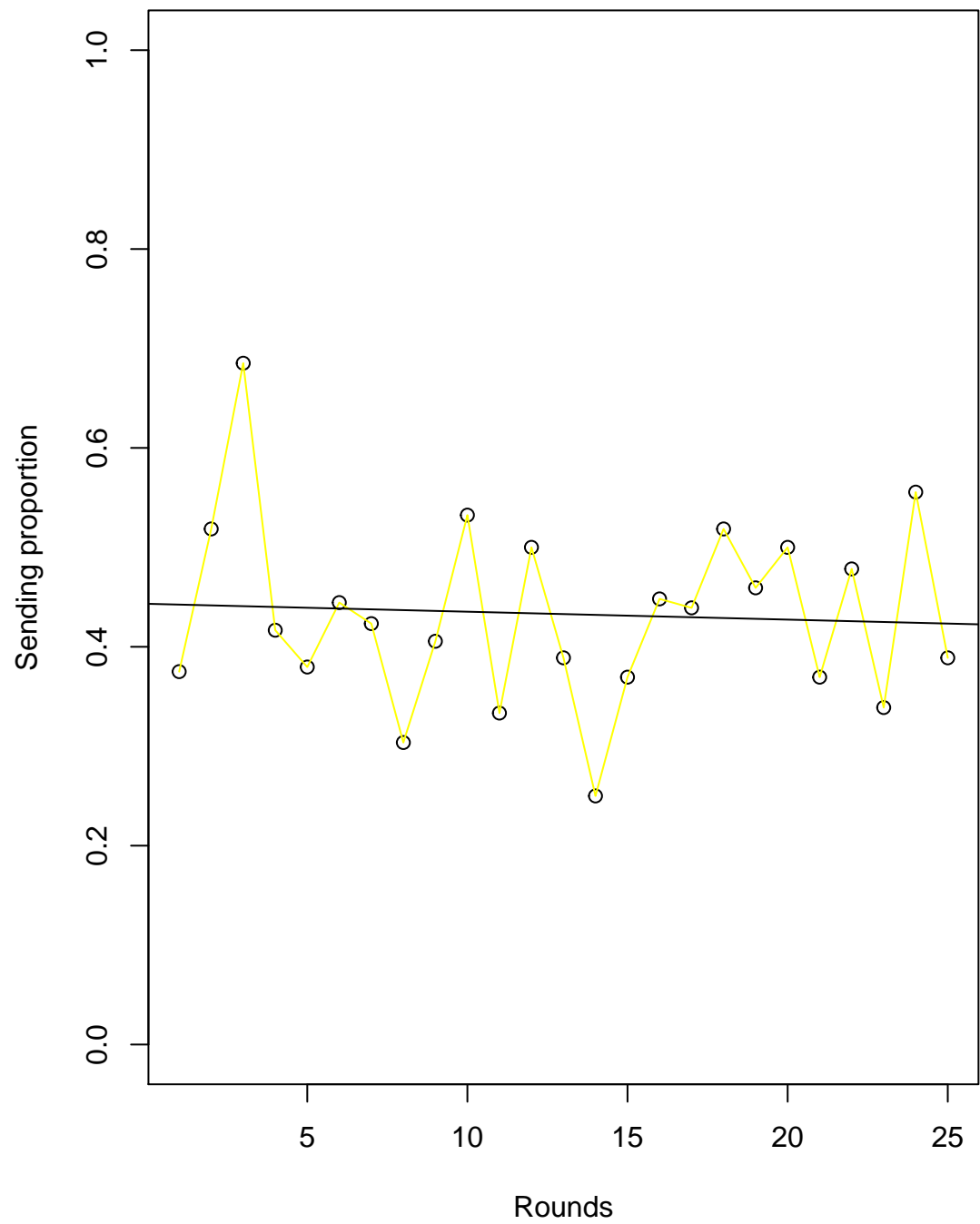
##  
##  
## \pagebreak

## Sending behavior of Combine Games over time of: SENDER

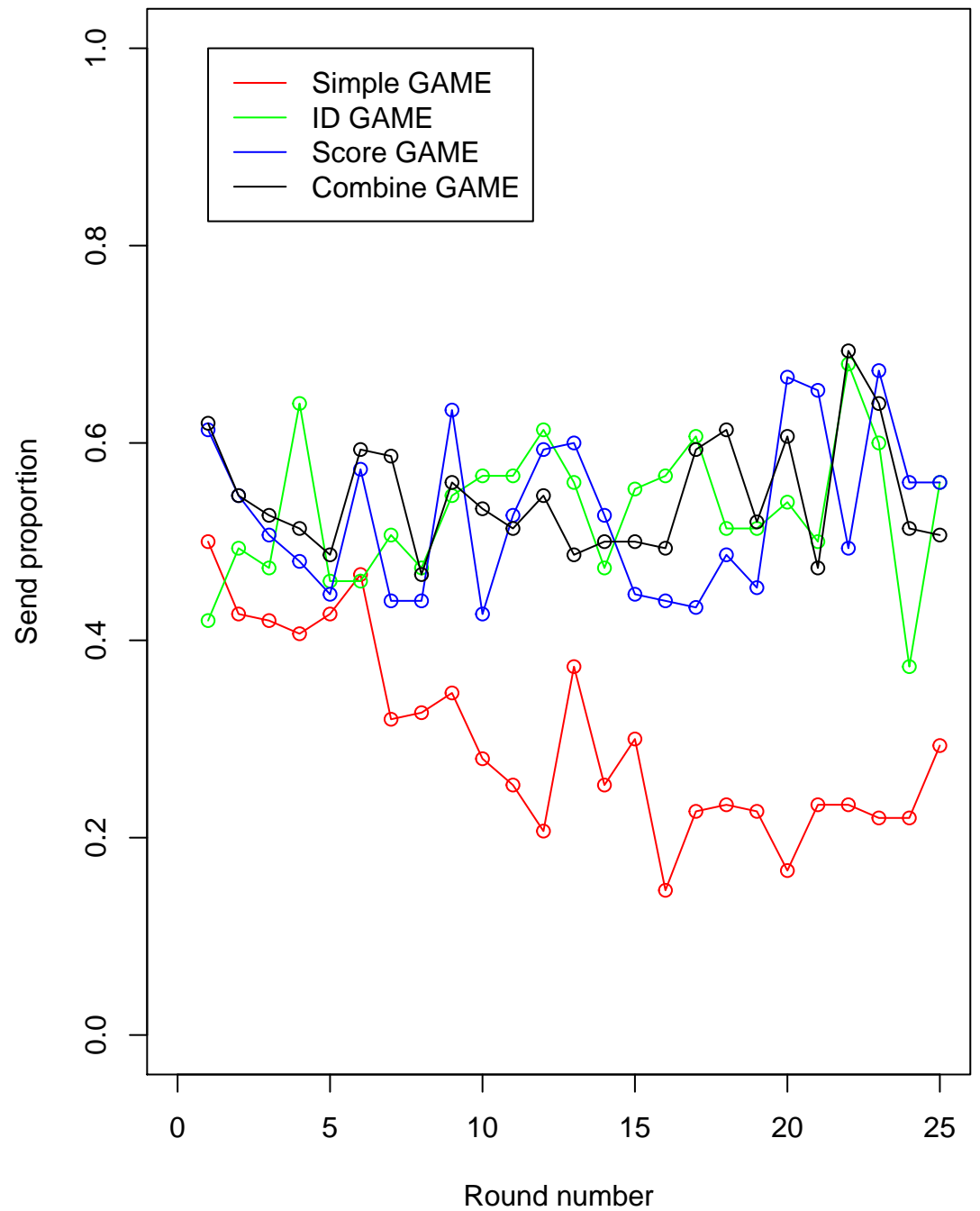


##  
##  
## \pagebreak

**Sending behavior of Combine Games over time of: RECEIVER**



##  
##  
## \pagebreak



date: Wed 7-Jun-2017

```
## [1] "GAME: Simple GAME for type: SENDER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12192 -0.03637  0.01025  0.02810  0.11977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept)  0.437533    0.024477  17.875 5.51e-15 ***
## rounds      -0.010559    0.001647  -6.413 1.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05937 on 23 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6257
## F-statistic: 41.13 on 1 and 23 DF,  p-value: 1.522e-06
##
## [1] "GAME: ID GAME  for type: SENDER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.183974 -0.044836 -0.007523  0.041159  0.131615
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.498600    0.028607  17.429 9.48e-15 ***
## rounds       0.002446    0.001924   1.271  0.216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06938 on 23 degrees of freedom
## Multiple R-squared:  0.06565, Adjusted R-squared:  0.02502
## F-statistic: 1.616 on 1 and 23 DF,  p-value: 0.2164
##
## [1] "GAME: Score GAME  for type: SENDER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.104472 -0.075292  0.000379  0.066785  0.122108
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.499533    0.033150  15.07 2.08e-13 ***
## rounds       0.002251    0.002230   1.01  0.323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0804 on 23 degrees of freedom
## Multiple R-squared:  0.04244, Adjusted R-squared:  0.0008026
## F-statistic: 1.019 on 1 and 23 DF,  p-value: 0.3232
##
## [1] "GAME: Combine GAME  for type: SENDER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##

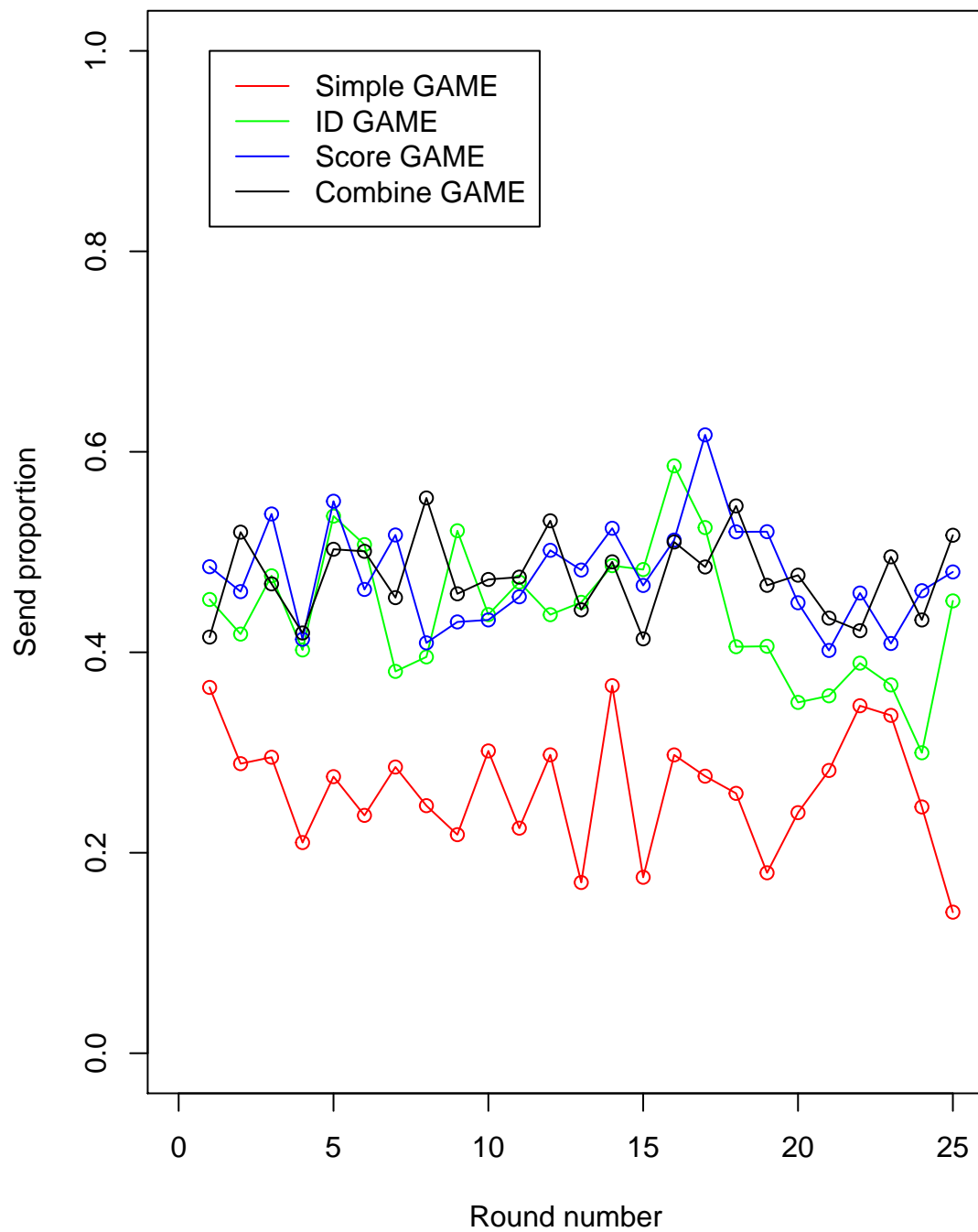
```

```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.079918 -0.047313 -0.009031  0.047272  0.139092
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.5324667  0.0244471  21.780  <2e-16 ***
## rounds      0.0009897  0.0016445   0.602    0.553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05929 on 23 degrees of freedom
## Multiple R-squared:  0.0155, Adjusted R-squared:  -0.0273
## F-statistic: 0.3622 on 1 and 23 DF,  p-value: 0.5532
##
##
## \pagebreak

```





```
## [1] "GAME: Simple GAME for type: RECEIVER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.104956 -0.040882  0.003702  0.033675  0.105474
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept)  0.280915    0.024914  11.276 7.58e-11 ***
## rounds      -0.001409    0.001676  -0.841    0.409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06042 on 23 degrees of freedom
## Multiple R-squared:  0.02981,    Adjusted R-squared:  -0.01238
## F-statistic: 0.7066 on 1 and 23 DF,  p-value: 0.4092
##
## [1] "GAME: ID GAME for type: RECEIVER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10325 -0.05641 -0.01202  0.04958  0.15635
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.482910    0.026250  18.396 2.97e-15 ***
## rounds      -0.003330    0.001766  -1.886    0.072 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06367 on 23 degrees of freedom
## Multiple R-squared:  0.1339, Adjusted R-squared:  0.09629
## F-statistic: 3.557 on 1 and 23 DF,  p-value: 0.07198
##
## [1] "GAME: Score GAME for type: RECEIVER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.07276 -0.02567 -0.01075  0.03587  0.14044
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.484435    0.021645  22.381 <2e-16 ***
## rounds      -0.000468    0.001456  -0.321    0.751
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0525 on 23 degrees of freedom
## Multiple R-squared:  0.004473,    Adjusted R-squared:  -0.03881
## F-statistic: 0.1033 on 1 and 23 DF,  p-value: 0.7508
##
## [1] "GAME: Combine GAME for type: RECEIVER"
##
## Call:
## lm(formula = send_behavior ~ rounds)
##

```

```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.062590 -0.033791 -0.001223  0.026182  0.077559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.769e-01  1.726e-02  27.634  <2e-16 ***
## rounds      -6.216e-05  1.161e-03  -0.054    0.958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04186 on 23 degrees of freedom
## Multiple R-squared:  0.0001246, Adjusted R-squared:  -0.04335
## F-statistic: 0.002867 on 1 and 23 DF, p-value: 0.9578
##
##
## \pagebreak

```

## Standard deviation of each user by game

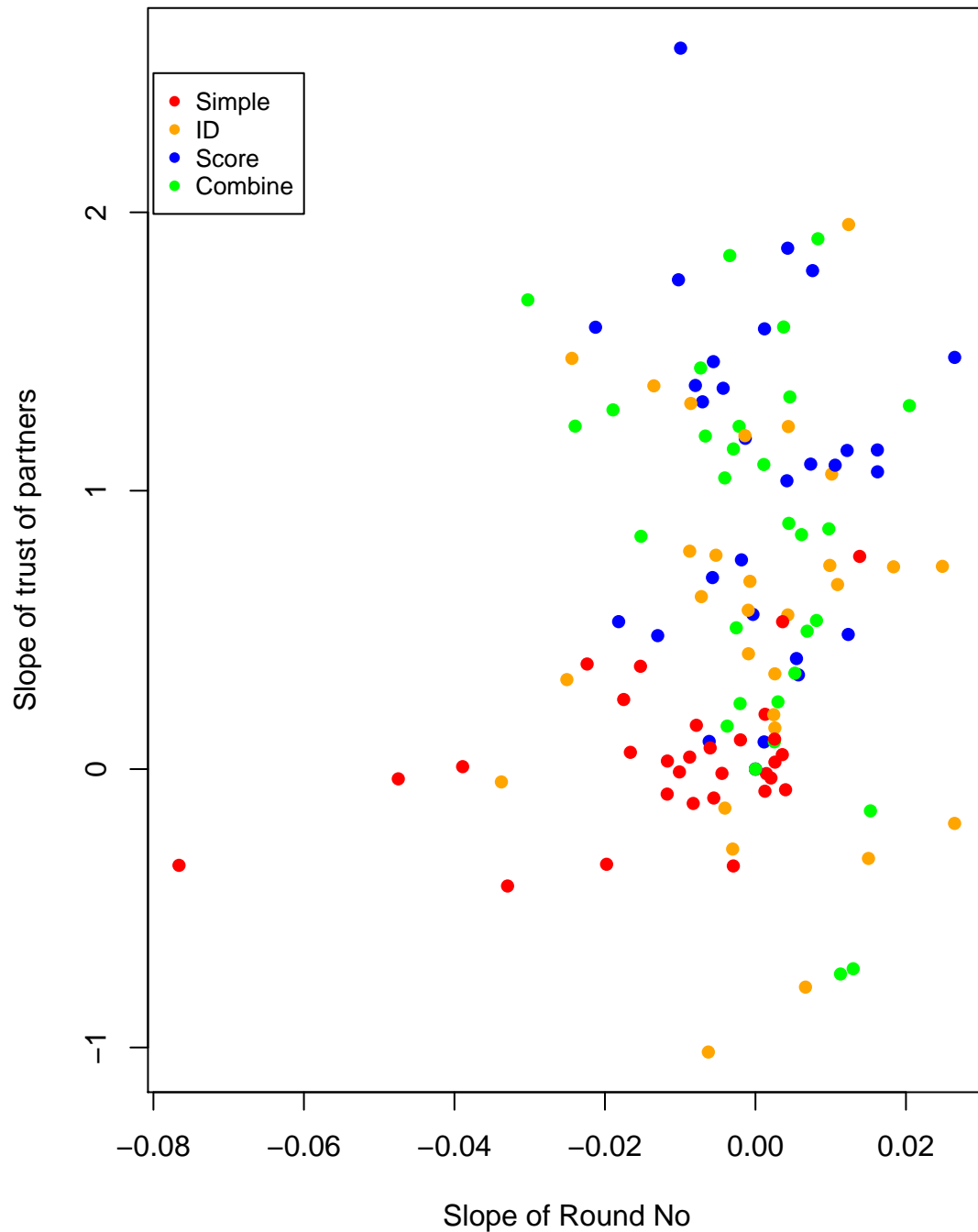
```
## [1] "Anova analysis for standard deviation of relative sending"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##              SS num Df Error SS den Df      F
## (Intercept)      6.2513      1 0.50229      25 311.1424
## GroupID          0.1419      4 0.50229      25  1.7659
## SHOW_TRUST       0.0233      1 0.24129      25  0.4137
## GroupID:SHOW_TRUST 0.2255      4 0.24129      25  5.8408
## SHOW_ID          0.0127      1 0.16420      25  0.8293
## GroupID:SHOW_ID   0.0610      4 0.16420      25  2.3232
## SHOW_TRUST:SHOW_ID 0.0547      1 0.17377      25  2.4027
## GroupID:SHOW_TRUST:SHOW_ID 0.0911      4 0.17377      25  3.2780
##
##              Pr(>F)
## (Intercept)      1.279e-15 ***
## GroupID          0.167331
## SHOW_TRUST       0.555113
## GroupID:SHOW_TRUST 0.001842 **
## SHOW_ID          0.413987
## GroupID:SHOW_ID   0.084418 .
## SHOW_TRUST:SHOW_ID 0.196058
## GroupID:SHOW_TRUST:SHOW_ID 0.027247 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*****--"
## [1] "Mean of standard deviation by game"
## [1] 0.182673
## [1] 0.253273
## [1] 0.245929
## [1] 0.231093
## [1] "Anova analysis for standard deviation of relative sending"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Receiver"
##
##              SS num Df Error SS den Df      F
## (Intercept)      2.52343      1 0.54722      25 115.2847
## GroupID          0.03184      4 0.54722      25  0.3636
## SHOW_TRUST       0.00407      1 0.21234      25  0.8069
## GroupID:SHOW_TRUST 0.02017      4 0.21234      25  0.5938
## SHOW_ID          0.00028      1 0.19251      25  0.0378
## GroupID:SHOW_ID   0.03008      4 0.19251      25  0.9767
## SHOW_TRUST:SHOW_ID 0.00268      1 0.11743      25  0.2861
## GroupID:SHOW_TRUST:SHOW_ID 0.03753      4 0.11743      25  1.9976
##
##              Pr(>F)
## (Intercept)      7.489e-11 ***
## GroupID          0.8321
## SHOW_TRUST       0.4198
## GroupID:SHOW_TRUST 0.6703
## SHOW_ID          0.8553
## GroupID:SHOW_ID   0.4379
## SHOW_TRUST:SHOW_ID 0.6211
## GroupID:SHOW_TRUST:SHOW_ID 0.1257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] "----****--"
## [1] "Mean of standard deviation by game"
## [1] 0.1359977
## [1] 0.1571053
## [1] 0.1423797
## [1] 0.144567
```

## Behavior on trust score over time

We want to see the adaptation of user on trust score over time.

### Regression on trust over time of SENDER



```
##      X.metric.....as.character.. X.criteria.....as.character..  
## 1              Slope 1              Without trust  
## 2              Slope 1              With trust  
## 3              Slope 2              Without trust  
## 4              Slope 2              With trust
```

```

## 5          Intercept          Without trust
## 6          Intercept          With trust
## 7          r_value          Without trust
## 8          r_value          With trust
## 9          r_value          Without ID
## 10         r_value          With ID
## X.Mean.....as.numeric.. X.std.....as.numeric..
## 1          0.01466          0.00729
## 2          0.01028          0.00705
## 3          0.62572          0.53448
## 4          1.20593          0.5589
## 5          0.5651          0.36383
## 6          0.19369          0.09538
## 7          0.5734          0.1638
## 8          0.53229          0.15817
## 9          0.59957          0.13413
## 10         0.53          0.17
## [1] "Anova analysis for slope of round ID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
##          SS num Df  Error SS den Df      F
## (Intercept)      0.00007475      1 0.0027016      24 0.6641
## GroupID          0.00078933      4 0.0027016      24 1.7530
## SHOW_TRUST       0.00043572      1 0.0029410      24 3.1151
## GroupID:SHOW_TRUST 0.00055950      4 0.0029410      24 1.1414
## SHOW_ID          0.00043619      1 0.0021703      24 5.9448
## GroupID:SHOW_ID   0.00029349      4 0.0021703      24 0.8114
## SHOW_TRUST:SHOW_ID 0.00092180      1 0.0018506      24 4.9127
## GroupID:SHOW_TRUST:SHOW_ID 0.00075054      4 0.0018506      24 2.4334
##
##          Pr(>F)
## (Intercept)      0.42314
## GroupID          0.17137
## SHOW_TRUST       0.15233
## GroupID:SHOW_TRUST 0.36098
## SHOW_ID          0.07135 .
## GroupID:SHOW_ID   0.53027
## SHOW_TRUST:SHOW_ID 0.09097 .
## GroupID:SHOW_TRUST:SHOW_ID 0.07505 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*****----"
## [1] "Anova analysis for slope of trust of partners"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Sender"
##
##          SS num Df Error SS den Df      F      Pr(>F)
## (Intercept)      8.9533      1 13.3546      24 16.0903 0.0005121
## GroupID          1.2310      4 13.3546      24 0.5531 0.6986806
## SHOW_TRUST       2.9256      1  6.2610      24 7.7710 0.0494306
## GroupID:SHOW_TRUST 1.5059      4  6.2610      24 1.4431 0.2504663
## SHOW_ID          0.0311      1  3.0610      24 0.5357 0.5047749
## GroupID:SHOW_ID   0.2322      4  3.0610      24 0.4551 0.7677338
## SHOW_TRUST:SHOW_ID 1.4568      1  6.5879      24 9.0870 0.0393770
## GroupID:SHOW_TRUST:SHOW_ID 0.6413      4  6.5879      24 0.5840 0.6771880
##

```

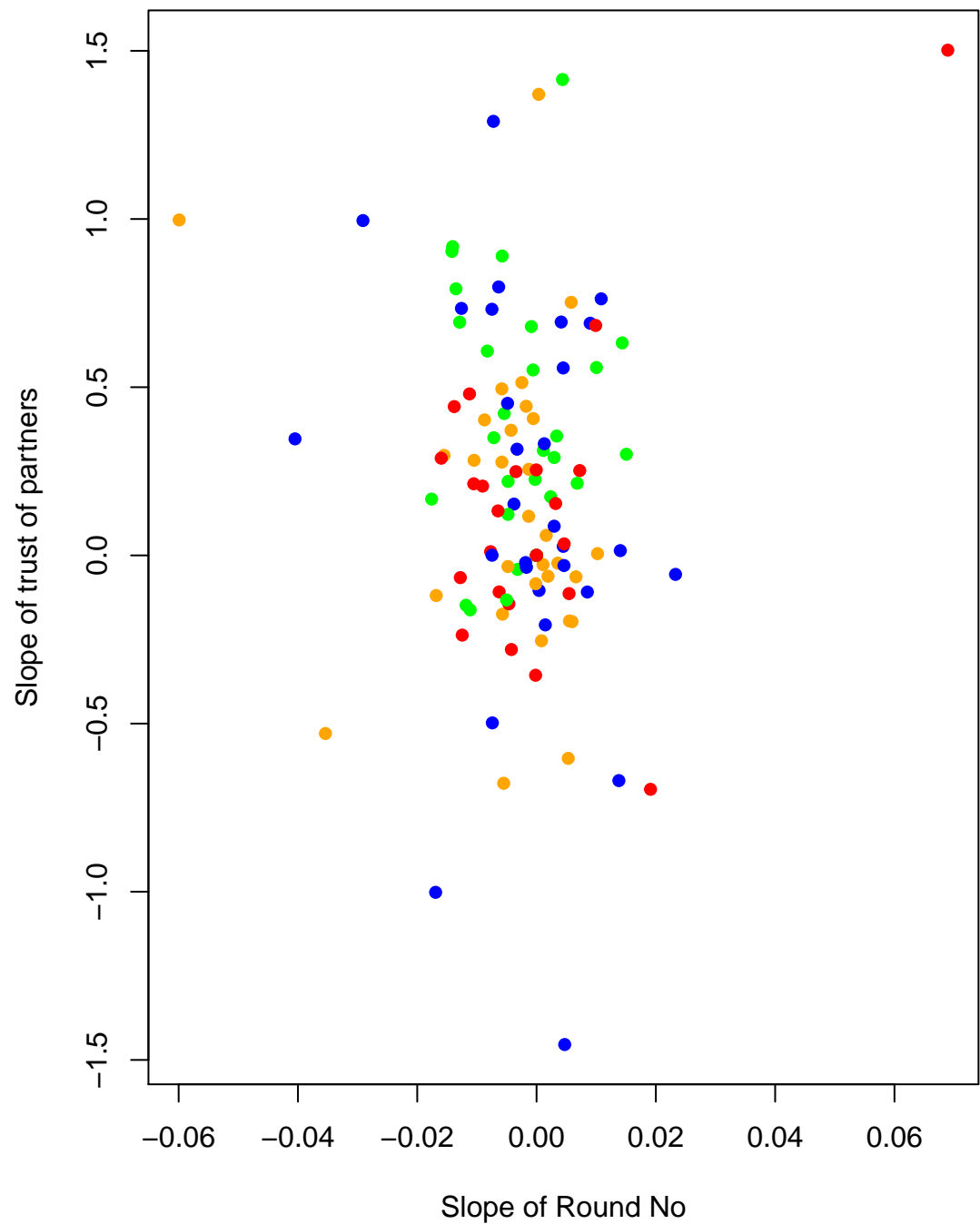
```

## (Intercept)          ***
## GroupID
## SHOW_TRUST           *
## GroupID:SHOW_TRUST
## SHOW_ID
## GroupID:SHOW_ID
## SHOW_TRUST:SHOW_ID   *
## GroupID:SHOW_TRUST:SHOW_ID
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "Anova analysis for intercept"
## [1] "-----*-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F  Pr(>F)
## (Intercept)      0.24865      1  1.63355      24 3.6531 0.06798 .
## GroupID          0.21266      4  1.63355      24 0.7811 0.54853
## SHOW_TRUST       0.50076      1  1.60492      24 9.3747 0.03759 *
## GroupID:SHOW_TRUST 0.21366      4  1.60492      24 0.7988 0.53781
## SHOW_ID          0.04311      1  0.97672      24 4.5564 0.09969 .
## GroupID:SHOW_ID   0.03785      4  0.97672      24 0.2325 0.91734
## SHOW_TRUST:SHOW_ID 0.24281      1  0.98892      24 9.8727 0.03478 *
## GroupID:SHOW_TRUST:SHOW_ID 0.09838      4  0.98892      24 0.5969 0.66838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"
## [1] "Anova analysis for R value"
## [1] "-----*-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##
##          SS num Df Error SS den Df      F  Pr(>F)
## (Intercept)      0.00098      1  2.1830      23 0.0103 0.919920
## GroupID          0.79302      4  2.1830      23 2.0888 0.115157
## SHOW_TRUST       0.25225      1  2.5088      23 3.7464 0.125014
## GroupID:SHOW_TRUST 0.26933      4  2.5088      23 0.6173 0.654617
## SHOW_ID          0.92531      1  1.9594      23 44.6771 0.002605
## GroupID:SHOW_ID   0.08284      4  1.9594      23 0.2431 0.910875
## SHOW_TRUST:SHOW_ID 0.32345      1  1.9235      23 1.8518 0.245195
## GroupID:SHOW_TRUST:SHOW_ID 0.69867      4  1.9235      23 2.0886 0.115188
##
## (Intercept)
## GroupID
## SHOW_TRUST
## GroupID:SHOW_TRUST
## SHOW_ID              **
## GroupID:SHOW_ID
## SHOW_TRUST:SHOW_ID
## GroupID:SHOW_TRUST:SHOW_ID
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "----*-----"

```



Regression on trust over time of RECEIVER



##	X.metric.....as.character..	X.criteria.....as.character..
## 1	Slope 1	Without trust
## 2	Slope 1	With trust
## 3	Slope 2	Without trust
## 4	Slope 2	With trust
## 5	Intercept	Without trust
## 6	Intercept	With trust
## 7	r_value	Without trust

```

## 8          r_value          With trust
## 9          r_value          Without ID
## 10         r_value          With ID
## X.Mean.....as.numeric.. X.std.....as.numeric..
## 1          0.03958          0.03939
## 2          0.00434          0.00268
## 3          0.67759          0.66709
## 4          0.48181          0.26954
## 5          0.76553          0.27779
## 6          0.461           0.16972
## 7          0.83548          0.02699
## 8          0.55454          0.04005
## 9          0.62422          0.12929
## 10         0.7             0.16
## [1] "Anova analysis for slope of round ID"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##          SS num Df  Error SS den Df      F
## (Intercept)      0.00005552      1 0.0010043      17 0.9398
## GroupID          0.00017305      4 0.0010043      17 0.7323
## SHOW_TRUST       0.00000227      1 0.0003877      17 0.0560
## GroupID:SHOW_TRUST 0.00016187      4 0.0003877      17 1.7742
## SHOW_ID          0.00021497      1 0.0023923      17 0.4144
## GroupID:SHOW_ID   0.00207482      4 0.0023923      17 3.6860
## SHOW_TRUST:SHOW_ID 0.00039331      1 0.0032947      17 0.9392
## GroupID:SHOW_TRUST:SHOW_ID 0.00167515      4 0.0032947      17 2.1609
##          Pr(>F)
## (Intercept)      0.3459
## GroupID          0.5824
## SHOW_TRUST       0.8246
## GroupID:SHOW_TRUST 0.1806
## SHOW_ID          0.5548
## GroupID:SHOW_ID   0.0245 *
## SHOW_TRUST:SHOW_ID 0.3874
## GroupID:SHOW_TRUST:SHOW_ID 0.1174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
## [1] "Anova analysis for slope of trust of partners"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##          SS num Df  Error SS den Df      F Pr(>F)
## (Intercept)      0.11002      1 0.77707      17 2.4070 0.13921
## GroupID          0.83202      4 0.77707      17 4.5505 0.01109 *
## SHOW_TRUST       0.01162      1 1.08190      17 0.0697 0.80480
## GroupID:SHOW_TRUST 0.66693      4 1.08190      17 2.6199 0.07158 .
## SHOW_ID          0.02173      1 0.19461      17 1.8777 0.24247
## GroupID:SHOW_ID   0.04628      4 0.19461      17 1.0107 0.42942
## SHOW_TRUST:SHOW_ID 0.01998      1 0.10908      17 0.8667 0.40456
## GroupID:SHOW_TRUST:SHOW_ID 0.09221      4 0.10908      17 3.5924 0.02680 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
## [1] "Anova analysis for intercept"

```

```
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##          SS num Df Error SS den Df      F Pr(>F)
## (Intercept)      0.33688      1  0.78376      17 7.3069 0.01508 *
## GroupID          0.11207      4  0.78376      17 0.6077 0.66254
## SHOW_TRUST       0.00018      1  0.41501      17 0.0023 0.96406
## GroupID:SHOW_TRUST 0.30754      4  0.41501      17 3.1494 0.04145 *
## SHOW_ID          0.07797      1  0.92954      17 0.4054 0.55893
## GroupID:SHOW_ID    0.76927      4  0.92954      17 3.5172 0.02882 *
## SHOW_TRUST:SHOW_ID 0.32187      1  1.67419      17 4.4288 0.10313
## GroupID:SHOW_TRUST:SHOW_ID 0.29070      4  1.67419      17 0.7380 0.57889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
## [1] "Anova analysis for R value"
## [1] "-----*****-----"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms): Receiver"
##
##          SS num Df Error SS den Df      F Pr(>F)
## (Intercept)      0.00241      1  1.28537      17 0.0318 0.86049
## GroupID          0.25044      4  1.28537      17 0.8281 0.52548
## SHOW_TRUST       0.03310      1  0.53545      17 0.4338 0.54610
## GroupID:SHOW_TRUST 0.30514      4  0.53545      17 2.4220 0.08838 .
## SHOW_ID          0.06071      1  0.75648      17 0.4218 0.55144
## GroupID:SHOW_ID    0.57568      4  0.75648      17 3.2342 0.03808 *
## SHOW_TRUST:SHOW_ID 0.00425      1  1.92036      17 0.1279 0.73868
## GroupID:SHOW_TRUST:SHOW_ID 0.13279      4  1.92036      17 0.2939 0.87788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "-----*****-----"
```

## Questionnaire analysis

In this section, we present the summary of questionnaire we asked participants after the experiment

```
print("What game the receivers will send back most, with the same amount of sending first?")

## [1] "What game the receivers will send back most, with the same amount of sending first?"
print(summary(SBJs$receive_back_most))

##               Game 1 (Simple Game)
##                               3
##           Game 3 (Partner Information Game)
##                               9
## Game 4 (Partner Identity and Information Game)
##                               13
##           Game 2 (Partner Identity Game)
##                               4
##           No idea / Do not remember
##                               1
print("What game is best for personal earning?")

## [1] "What game is best for personal earning?"
print(summary(SBJs$best_personal))

##           Game 3 (Partner Information Game)
##                               9
## Game 4 (Partner Identity and Information Game)
##                               16
##           Game 1 (Simple Game)
##                               2
##           Game 2 (Partner Identity Game)
##                               3
print("What game is worst for personal earning?")

## [1] "What game is worst for personal earning?"
print(summary(SBJs$worst_personal))

##               Game 1 (Simple Game)
##                               26
## Game 4 (Partner Identity and Information Game)
##                               3
##           Game 3 (Partner Information Game)
##                               1
print("What game is best for total earning?")

## [1] "What game is best for total earning?"
print(summary(SBJs$best_total))

##           Game 3 (Partner Information Game)
##                               10
## Game 4 (Partner Identity and Information Game)
##                               15
```

```

##                No idea / Do not remember
##                1
##                Game 1 (Simple Game)
##                1
##                Game 2 (Partner Identity Game)
##                3
print("What game is worst for total earning?")

## [1] "What game is worst for total earning?"
print(summary(SBJs$worst_total))

##                Game 1 (Simple Game)
##                24
##                Game 2 (Partner Identity Game)
##                2
## Game 4 (Partner Identity and Information Game)
##                2
##                No idea / Do not remember
##                1
##                Game 3 (Partner Information Game)
##                1
print("In Simple Game, profit is higher if you send more?")

## [1] "In Simple Game, profit is higher if you send more?"
print(summary(SBJs$send_more_for_profit))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.0    1.0    1.0    1.8    3.0    4.0
print("In Simple Game, profit is higher if you send less?")

## [1] "In Simple Game, profit is higher if you send less?"
print(summary(SBJs$send_less_for_profit))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.000    2.000    2.500    2.633    3.750    4.000
print("In Simple Game, you are receive, you send back more if your sender send more?")

## [1] "In Simple Game, you are receive, you send back more if your sender send more?"
print(summary(SBJs$trust_help_receiver))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.000    2.000    3.000    2.733    4.000    5.000
print("Show ID and Score help to realize behavior of partners in the history?")

## [1] "Show ID and Score help to realize behavior of partners in the history?"
print(summary(SBJs$show_id_help))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.000    2.000    3.000    2.633    3.000    5.000
print(summary(SBJs$show_score_help))

```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      3.0      3.0      3.2      4.0      5.0
print(summary(SBJs$show_combine_help))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.000    2.250    3.000    2.667    3.000    4.000
print("In Combine game, trust score reflects correct behavior of the partner?")

## [1] "In Combine game, trust score reflects correct behavior of the partner?"
print(summary(SBJs$trust_score_correctness))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.000    3.000    3.000    3.267    4.000    5.000
print("In Score game, you send more if your partner has higher trust score?")

## [1] "In Score game, you send more if your partner has higher trust score?"
print(summary(SBJs$trust_help_sender))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      2.0      3.0      2.8      3.0      5.0
print("Showing ID and score help you decide how to behave?")

## [1] "Showing ID and score help you decide how to behave?"
print(summary(SBJs$identity_help_decide))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.000    2.000    3.000    2.633    3.000    5.000
print(summary(SBJs$trust_score_help_decide))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      3.0      3.0      3.2      4.0      4.0
print(summary(SBJs$combine_help_decide))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      3.0      4.0      3.3      4.0      4.0
print("What factor is more important?")

## [1] "What factor is more important?"
print(summary(SBJs$important_factor))

##      Identity No Preference    Trust score
##      16          1          13
print("Do you think your partners are fair?")

## [1] "Do you think your partners are fair?"
print(summary(SBJs$partner_fair))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      2.0      2.0      2.4      3.0      5.0
```

## Chi square test between factorial questions

We want to see if the questionnaire showed the correct experience of users.

For factorial question, we will calculate chi - square test to see whether they are significant or not.

```
chi_square(SBJs$receive_back_most, SBJs$best_personal)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    3    0    0    0
## [2,]    4    4    0    1
## [3,]    2    9    1    1
## [4,]    0    2    1    1
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.08833
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$receive_back_most, SBJs$worst_personal)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    1    0    0
## [2,]    9    0    0    0
## [3,]   11    2    0    0
## [4,]    3    0    1    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.173
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$receive_back_most, SBJs$best_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    2    1    0
## [2,]    4    4    0    0
## [3,]    4    8    0    1
## [4,]    2    0    0    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.187
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$receive_back_most, SBJs$worst_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    1    0    0
## [2,]    6    1    1    1
## [3,]   12    0    1    0
## [4,]    3    0    0    0
##
## Fisher's Exact Test for Count Data
```

```
##
## data: mm
## p-value = 0.457
## alternative hypothesis: two.sided
chi_square(SBJs$best_personal, SBJs$worst_personal)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   8   1   0   0
## [2,]  16   0   0   0
## [3,]   0   1   1   0
## [4,]   2   1   0   0
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.003941
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$best_personal, SBJs$best_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   4   4   1   0
## [2,]   5  10   0   0
## [3,]   1   0   0   1
## [4,]   0   1   0   0
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.1626
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$best_personal, SBJs$worst_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   6   2   0   1
## [2,]  14   0   1   0
## [3,]   1   0   1   0
## [4,]   3   0   0   0
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.1317
## alternative hypothesis: two.sided
```

```
chi_square(SBJs$worst_personal, SBJs$best_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   9  14   0   0
## [2,]   0   1   1   1
## [3,]   1   0   0   0
## [4,]   0   0   0   0
##
## Fisher's Exact Test for Count Data
##
```



```
## data: mm
## p-value = 0.01752
## alternative hypothesis: two.sided
chi_square(SBJs$worst_personal, SBJs$worst_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  22   1   1   1
## [2,]   1   1   1   0
## [3,]   1   0   0   0
## [4,]   0   0   0   0
##
## Fisher's Exact Test for Count Data
##
```

```
## data: mm
## p-value = 0.1478
## alternative hypothesis: two.sided
chi_square(SBJs$best_total, SBJs$worst_total)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   8   0   1   1
## [2,]  14   1   0   0
## [3,]   0   1   0   0
## [4,]   0   0   1   0
##
## Fisher's Exact Test for Count Data
##
```

```
## data: mm
## p-value = 0.01349
## alternative hypothesis: two.sided
```

## Comparing questionnaire with real data

### Highest sending back

In the first question, we ask what game the receivers will send back most, given the same amount of sending first by senders.

First, we analyze for all games. From the above analysis (basic data analysis), we know the increasing order is: Simple Game < Score Game < ID Game < Combine Game, but actually the difference between 3 last games are very small.

### Best and worst game for personal earning and total earning

In the questionnaire, for the best game, 16 people selected game 4, 9 selected game 3 (Score Game), 3 and 2 selected Game 2 (ID Game) and Game 1 (Simple Game) respectively.

For the worst game, the numbers are: 26 for game 1, 1 for Game 3 and 3 for Game 4.

For best game for total earnings,

We can analyze the data to see what is the correct answer, the numbers selected Game 1, 2, 3, 4 are 1, 3, 10, 15 respectively, and there is 1 person has no idea

For the worst, the numbers are 24, 2, 1, 2, and again there is 1 person has no idea.

```
best_personal_earnings = c(0, 0, 0, 0)
worst_personal_earnings = c(0, 0, 0, 0)
best_total_earnings = c(0, 0, 0, 0)
worst_total_earnings = c(0, 0, 0, 0)

real_best_personal = as.numeric()
real_worst_personal = as.numeric()
real_best_total = as.numeric()
real_worst_total = as.numeric()
for (exp_id in 1:num_exp) {
  first_round_of_exp_subjects = (exp_id - 1) * num_rounds_per_game *
    num_users + 1
  last_round_of_exp_subjects = exp_id * num_rounds_per_game *
    num_users

  simple_game = simple_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  id_game = id_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  score_game = score_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  combine_game = combine_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]

  for (user_id in 1:num_users) {
    earning = c(sum(simple_game[simple_game$Subject ==
      user_id, ]$CurrGameProfit), sum(id_game[id_game$Subject ==
      user_id, ]$CurrGameProfit), sum(score_game[score_game$Subject ==
      user_id, ]$CurrGameProfit), sum(combine_game[combine_game$Subject ==
      user_id, ]$CurrGameProfit))
    best_personal_earnings[which.max(earning)] = best_personal_earnings[which.max(earning)] +
```

```

        1
        worst_personal_earnings[which.min(earning)] = worst_personal_earnings[which.min(earning)] +
        1
        real_best_personal <- c(real_best_personal,
                                which.max(earning))
        real_worst_personal <- c(real_worst_personal,
                                which.min(earning))
    }
    earning = c(sum(simple_game$CurrGameProfit), sum(id_game$CurrGameProfit),
                sum(score_game$CurrGameProfit), sum(combine_game$CurrGameProfit))
    best_total_earnings[which.max(earning)] = best_total_earnings[which.max(earning)] +
    1
    worst_total_earnings[which.min(earning)] = worst_total_earnings[which.min(earning)] +
    1

    real_best_total <- c(real_best_total, which.max(earning))
    real_worst_total <- c(real_worst_total, which.min(earning))
}
print(best_personal_earnings)

## [1] 3 9 7 11
print(worst_personal_earnings)

## [1] 25 3 2 0
print(best_total_earnings)

## [1] 0 3 0 2
print(worst_total_earnings)

## [1] 4 0 1 0
print("Comparing questionnaire with real data, one by one")

## [1] "Comparing questionnaire with real data, one by one"
print(data.frame(SBJs$best_personal, real_best_personal))

##
##          SBJs.best_personal real_best_personal
## 1 Game 4 (Partner Identity and Information Game)      3
## 2          Game 3 (Partner Information Game)      2
## 3 Game 4 (Partner Identity and Information Game)      2
## 4 Game 4 (Partner Identity and Information Game)      3
## 5 Game 4 (Partner Identity and Information Game)      3
## 6          Game 3 (Partner Information Game)      2
## 7          Game 2 (Partner Identity Game)      2
## 8 Game 4 (Partner Identity and Information Game)      4
## 9          Game 1 (Simple Game)      4
## 10         Game 3 (Partner Information Game)      4
## 11         Game 2 (Partner Identity Game)      2
## 12         Game 3 (Partner Information Game)      3
## 13 Game 4 (Partner Identity and Information Game)      4
## 14 Game 4 (Partner Identity and Information Game)      4
## 15         Game 3 (Partner Information Game)      4
## 16 Game 4 (Partner Identity and Information Game)      4

```

```
## 17 Game 4 (Partner Identity and Information Game) 3
## 18 Game 4 (Partner Identity and Information Game) 4
## 19 Game 2 (Partner Identity Game) 3
## 20 Game 4 (Partner Identity and Information Game) 2
## 21 Game 3 (Partner Information Game) 4
## 22 Game 4 (Partner Identity and Information Game) 2
## 23 Game 4 (Partner Identity and Information Game) 2
## 24 Game 4 (Partner Identity and Information Game) 4
## 25 Game 3 (Partner Information Game) 3
## 26 Game 3 (Partner Information Game) 1
## 27 Game 3 (Partner Information Game) 1
## 28 Game 1 (Simple Game) 1
## 29 Game 4 (Partner Identity and Information Game) 2
## 30 Game 4 (Partner Identity and Information Game) 4
```

```
print(data.frame(SBJs$worst_personal, real_worst_personal))
```

```
## SBJs.worst_personal real_worst_personal
## 1 Game 1 (Simple Game) 1
## 2 Game 4 (Partner Identity and Information Game) 1
## 3 Game 1 (Simple Game) 1
## 4 Game 1 (Simple Game) 1
## 5 Game 1 (Simple Game) 1
## 6 Game 1 (Simple Game) 1
## 7 Game 1 (Simple Game) 1
## 8 Game 1 (Simple Game) 1
## 9 Game 4 (Partner Identity and Information Game) 1
## 10 Game 1 (Simple Game) 1
## 11 Game 1 (Simple Game) 1
## 12 Game 1 (Simple Game) 1
## 13 Game 1 (Simple Game) 1
## 14 Game 1 (Simple Game) 1
## 15 Game 1 (Simple Game) 1
## 16 Game 1 (Simple Game) 1
## 17 Game 1 (Simple Game) 1
## 18 Game 1 (Simple Game) 2
## 19 Game 4 (Partner Identity and Information Game) 1
## 20 Game 1 (Simple Game) 1
## 21 Game 1 (Simple Game) 1
## 22 Game 1 (Simple Game) 1
## 23 Game 1 (Simple Game) 1
## 24 Game 1 (Simple Game) 1
## 25 Game 1 (Simple Game) 2
## 26 Game 1 (Simple Game) 3
## 27 Game 1 (Simple Game) 2
## 28 Game 3 (Partner Information Game) 3
## 29 Game 1 (Simple Game) 1
## 30 Game 1 (Simple Game) 1
```

```
print(data.frame(SBJs$best_total, real_best_total))
```

```
## SBJs.best_total real_best_total
## 1 Game 4 (Partner Identity and Information Game) 2
## 2 No idea / Do not remember 2
## 3 Game 4 (Partner Identity and Information Game) 4
```

## 4	Game 3 (Partner Information Game)	2
## 5	Game 4 (Partner Identity and Information Game)	4
## 6	Game 4 (Partner Identity and Information Game)	2
## 7	Game 2 (Partner Identity Game)	2
## 8	Game 3 (Partner Information Game)	4
## 9	Game 1 (Simple Game)	2
## 10	Game 3 (Partner Information Game)	4
## 11	Game 2 (Partner Identity Game)	2
## 12	Game 4 (Partner Identity and Information Game)	2
## 13	Game 4 (Partner Identity and Information Game)	4
## 14	Game 4 (Partner Identity and Information Game)	2
## 15	Game 3 (Partner Information Game)	4
## 16	Game 4 (Partner Identity and Information Game)	2
## 17	Game 4 (Partner Identity and Information Game)	2
## 18	Game 4 (Partner Identity and Information Game)	4
## 19	Game 4 (Partner Identity and Information Game)	2
## 20	Game 2 (Partner Identity Game)	4
## 21	Game 3 (Partner Information Game)	2
## 22	Game 3 (Partner Information Game)	2
## 23	Game 4 (Partner Identity and Information Game)	4
## 24	Game 3 (Partner Information Game)	2
## 25	Game 3 (Partner Information Game)	4
## 26	Game 4 (Partner Identity and Information Game)	2
## 27	Game 4 (Partner Identity and Information Game)	2
## 28	Game 3 (Partner Information Game)	4
## 29	Game 3 (Partner Information Game)	2
## 30	Game 4 (Partner Identity and Information Game)	4

```
print(data.frame(SBJs$worst_total, real_worst_total))
```

##	SBJs.worst_total	real_worst_total
## 1	Game 1 (Simple Game)	1
## 2	Game 2 (Partner Identity Game)	1
## 3	Game 1 (Simple Game)	1
## 4	Game 1 (Simple Game)	1
## 5	Game 1 (Simple Game)	3
## 6	Game 1 (Simple Game)	1
## 7	Game 1 (Simple Game)	1
## 8	Game 1 (Simple Game)	1
## 9	Game 4 (Partner Identity and Information Game)	1
## 10	No idea / Do not remember	3
## 11	Game 1 (Simple Game)	1
## 12	Game 1 (Simple Game)	1
## 13	Game 1 (Simple Game)	1
## 14	Game 1 (Simple Game)	1
## 15	Game 1 (Simple Game)	3
## 16	Game 1 (Simple Game)	1
## 17	Game 1 (Simple Game)	1
## 18	Game 1 (Simple Game)	1
## 19	Game 1 (Simple Game)	1
## 20	Game 3 (Partner Information Game)	3
## 21	Game 1 (Simple Game)	1
## 22	Game 1 (Simple Game)	1
## 23	Game 1 (Simple Game)	1
## 24	Game 4 (Partner Identity and Information Game)	1

## 25	Game 1 (Simple Game)	3
## 26	Game 2 (Partner Identity Game)	1
## 27	Game 1 (Simple Game)	1
## 28	Game 1 (Simple Game)	1
## 29	Game 1 (Simple Game)	1
## 30	Game 1 (Simple Game)	3

## Consistency between questionnaire and real data in best personal earning

```
# profit of game people believe that it is the best
best_person_earning_questionnaire = as.numeric()
worst_person_earning_questionnaire = as.numeric()

# profit of game which is really best for personal
# earning
real_best_person_earning = as.numeric()
real_worst_person_earning = as.numeric()

for (exp_id in 1:num_exp) {
  first_round_of_exp_subjects = (exp_id - 1) * num_rounds_per_game *
    num_users + 1
  last_round_of_exp_subjects = exp_id * num_rounds_per_game *
    num_users

  simple_game = simple_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  id_game = id_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  score_game = score_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  combine_game = combine_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]

  for (user_id in 1:num_users) {
    earning = c(sum(simple_game[simple_game$Subject ==
      user_id, ]$CurrGameProfit), sum(id_game[id_game$Subject ==
      user_id, ]$CurrGameProfit), sum(score_game[score_game$Subject ==
      user_id, ]$CurrGameProfit), sum(combine_game[combine_game$Subject ==
      user_id, ]$CurrGameProfit))

    best_person_earning_questionnaire <- c(best_person_earning_questionnaire,
      earning[as.numeric(SBJs$best_personal[(user_id +
        (exp_id - 1) * num_users))])])
    real_best_person_earning <- c(real_best_person_earning,
      max(earning))

    worst_person_earning_questionnaire <- c(worst_person_earning_questionnaire,
      earning[as.numeric(SBJs$worst_personal[(user_id +
        (exp_id - 1) * num_users))])])
    real_worst_person_earning <- c(real_worst_person_earning,
      min(earning))
    # print ('---') print
    # (earning[as.numeric(SBJs$worst_personal[(user_id
    # + (exp_id - 1) * num_users))])]) print
    # (min(earning))
  }
}

print(t.test(best_person_earning_questionnaire, real_best_person_earning,
  paired = TRUE))
```

```

##
## Paired t-test
##
## data: best_person_earning_questionnaire and real_best_person_earning
## t = -5.9516, df = 29, p-value = 1.819e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -77.08027 -37.65307
## sample estimates:
## mean of the differences
## -57.36667

print(t.test(worst_person_earning_questionnaire, real_worst_person_earning,
  paired = TRUE))

##
## Paired t-test
##
## data: worst_person_earning_questionnaire and real_worst_person_earning
## t = 2.3022, df = 29, p-value = 0.0287
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 1.748415 29.584918
## sample estimates:
## mean of the differences
## 15.66667

```



## Chi square test between questionnaire and real data

```
best_personal_earnings = c(0, 0, 0, 0)
worst_personal_earnings = c(0, 0, 0, 0)
best_total_earnings = c(0, 0, 0, 0)
worst_total_earnings = c(0, 0, 0, 0)

real_best_personal = as.numeric()
real_worst_personal = as.numeric()
real_best_total = as.numeric()
real_worst_total = as.numeric()
for (exp_id in 1:num_exp) {
  first_round_of_exp_subjects = (exp_id - 1) * num_rounds_per_game *
    num_users + 1
  last_round_of_exp_subjects = exp_id * num_rounds_per_game *
    num_users

  simple_game = simple_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  id_game = id_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  score_game = score_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  combine_game = combine_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]

  for (user_id in 1:num_users) {
    earning = c(sum(simple_game[simple_game$Subject ==
      user_id, ]$CurrGameProfit), sum(id_game[id_game$Subject ==
      user_id, ]$CurrGameProfit), sum(score_game[score_game$Subject ==
      user_id, ]$CurrGameProfit), sum(combine_game[combine_game$Subject ==
      user_id, ]$CurrGameProfit))
    best_personal_earnings[which.max(earning)] = best_personal_earnings[which.max(earning)] +
      1
    worst_personal_earnings[which.min(earning)] = worst_personal_earnings[which.min(earning)] +
      1
    real_best_personal <- c(real_best_personal,
      which.max(earning))
    real_worst_personal <- c(real_worst_personal,
      which.min(earning))
  }
  earning = c(sum(simple_game$CurrGameProfit), sum(id_game$CurrGameProfit),
    sum(score_game$CurrGameProfit), sum(combine_game$CurrGameProfit))
  best_total_earnings[which.max(earning)] = best_total_earnings[which.max(earning)] +
    1
  worst_total_earnings[which.min(earning)] = worst_total_earnings[which.min(earning)] +
    1

  # repeat 6 times for 6 users because all of 6 users
  # have the same best game for total earning
  for (i in 1:num_users) {
    real_best_total <- c(real_best_total, which.max(earning))
    real_worst_total <- c(real_worst_total, which.min(earning))
  }
}
```

```

    }
}

chi_square(SBJs$best_personal, real_best_personal)

##      [,1] [,2] [,3] [,4]
## [1,]    2    2    2    3
## [2,]    0    5    4    7
## [3,]    1    0    0    1
## [4,]    0    2    1    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.3075
## alternative hypothesis: two.sided

chi_square(SBJs$worst_personal, real_worst_personal)

##      [,1] [,2] [,3] [,4]
## [1,]   22    3    1    0
## [2,]    3    0    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.1607
## alternative hypothesis: two.sided

chi_square(SBJs$best_total, real_best_total)

##      [,1] [,2] [,3] [,4]
## [1,]    0    6    0    4
## [2,]    0    7    0    8
## [3,]    0    1    0    0
## [4,]    0    1    0    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm
## p-value = 0.9067
## alternative hypothesis: two.sided

chi_square(SBJs$worst_total, real_worst_total)

##      [,1] [,2] [,3] [,4]
## [1,]   19    0    5    0
## [2,]    1    0    1    0
## [3,]    2    0    0    0
## [4,]    1    0    0    0
##
## Fisher's Exact Test for Count Data
##
## data:  mm

```

```
## p-value = 0.7167
## alternative hypothesis: two.sided
```

## Chi square after reducing the dimension of game

Above, we analyze chi - square for each game (game 1, 2, 3, 4). In this section, we reduce the game to games: without and with trust, or without and with ID.

```
best_personal_earnings = c(0, 0, 0, 0)
worst_personal_earnings = c(0, 0, 0, 0)
best_total_earnings = c(0, 0, 0, 0)
worst_total_earnings = c(0, 0, 0, 0)

real_best_personal = as.numeric()
real_worst_personal = as.numeric()
real_best_total = as.numeric()
real_worst_total = as.numeric()
for (exp_id in 1:num_exp) {
  first_round_of_exp_subjects = (exp_id - 1) * num_rounds_per_game *
    num_users + 1
  last_round_of_exp_subjects = exp_id * num_rounds_per_game *
    num_users

  simple_game = simple_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  id_game = id_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  score_game = score_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]
  combine_game = combine_games[first_round_of_exp_subjects:last_round_of_exp_subjects,
    ]

  for (user_id in 1:num_users) {
    earning = c(sum(simple_game[simple_game$Subject ==
      user_id, ]$CurrGameProfit), sum(id_game[id_game$Subject ==
      user_id, ]$CurrGameProfit), sum(score_game[score_game$Subject ==
      user_id, ]$CurrGameProfit), sum(combine_game[combine_game$Subject ==
      user_id, ]$CurrGameProfit))
    best_personal_earnings[which.max(earning)] = best_personal_earnings[which.max(earning)] +
      1
    worst_personal_earnings[which.min(earning)] = worst_personal_earnings[which.min(earning)] +
      1
    real_best_personal <- c(real_best_personal,
      which.max(earning))
    real_worst_personal <- c(real_worst_personal,
      which.min(earning))
  }
  earning = c(sum(simple_game$CurrGameProfit), sum(id_game$CurrGameProfit),
    sum(score_game$CurrGameProfit), sum(combine_game$CurrGameProfit))
  best_total_earnings[which.max(earning)] = best_total_earnings[which.max(earning)] +
    1
  worst_total_earnings[which.min(earning)] = worst_total_earnings[which.min(earning)] +
    1

  # repeat 6 times for 6 users because all of 6 users
  # have the same best game for total earning
  for (i in 1:num_users) {
```

```

    real_best_total <- c(real_best_total, which.max(earning))
    real_worst_total <- c(real_worst_total, which.min(earning))
  }
}

```

```
print("Without and with trust")
```

```
## [1] "Without and with trust"
```

```
chi_square_2x2(floor(as.numeric(SBJs$best_personal)/2),
  floor(as.numeric(real_best_personal)/2))
```

```
##      [,1] [,2]
## [1,]   20   1
## [2,]    7   2
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.2069
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2430957 352.5438640
## sample estimates:
## odds ratio
##  5.332763

```

```
print("Without and with ID")
```

```
## [1] "Without and with ID"
```

```
chi_square_2x2(floor(as.numeric(SBJs$best_personal)%2),
  floor(as.numeric(real_best_personal)%2))
```

```
##      [,1] [,2]
## [1,]    5    6
## [2,]    5   14
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.4253
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.3686582 14.5597297
## sample estimates:
## odds ratio
##  2.264668

```

```
print("Without and with trust")
```

```
## [1] "Without and with trust"
```

```
chi_square_2x2(floor(as.numeric(SBJs$worst_personal)/2),
  floor(as.numeric(real_worst_personal)/2))
```

```
##      [,1] [,2]

```

```

## [1,] 1 3
## [2,] 4 22
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 0.5384
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.02800005 30.45845665
## sample estimates:
## odds ratio
## 1.790938

print("Without and with ID")

## [1] "Without and with ID"

chi_square_2x2(floor(as.numeric(SBJs$worst_personal)%2),
               floor(real_worst_personal%2))

##      [,1] [,2]
## [1,] 24 3
## [2,] 3 0
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.00000 27.89887
## sample estimates:
## odds ratio
## 0

print("Without and with trust")

## [1] "Without and with trust"

chi_square_2x2(floor(as.numeric(SBJs$best_total)/2),
               floor(real_best_total/2))

##      [,1] [,2]
## [1,] 20 0
## [2,] 10 0
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0 Inf
## sample estimates:
## odds ratio
## 0

```

```

print("Without and with ID")

## [1] "Without and with ID"
chi_square_2x2(floor(as.numeric(SBJs$best_total)%%2),
  floor(real_best_total%%2))

##      [,1] [,2]
## [1,]    0  14
## [2,]    0  16
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##      0 Inf
## sample estimates:
## odds ratio
##      0

print("Without and with trust")

## [1] "Without and with trust"
chi_square_2x2(floor(as.numeric(SBJs$worst_total)/2),
  floor(real_worst_total/2))

##      [,1] [,2]
## [1,]    1    5
## [2,]    5   19
##
## Fisher's Exact Test for Count Data
##
## data: mm
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0134252 9.6981212
## sample estimates:
## odds ratio
##  0.7666526

print("Without and with ID")

## [1] "Without and with ID"
chi_square_2x2(floor(as.numeric(SBJs$worst_total)%%2),
  floor(real_worst_total%%2))

##      [,1] [,2]
## [1,]   27    0
## [2,]    3    0
##
## Fisher's Exact Test for Count Data
##

```

```
## data:  mm
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##      0 Inf
## sample estimates:
## odds ratio
##          0
```