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 predicing 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 expriments from zTree and zLeaf 3.4.7.

There are 5 experiments so far, and both data files (XLS) and questionaire 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 interchangelly.

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))

# add epsilon value to avoid the case 0/0 EPSILON =
# 0.001 zTT[2]$subjects$my_send_proportional <-
# ifelse(zTT[2]$subjects$Type == 0,
# zTT[2]$subjects$Contribution/10,
# zTT[2]$subjects$Contribution/3/
# (zTT[2]$subjects$PartnerDecision + EPSILON))

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")

col_list <- c("red", "green", "blue", "black", "orange",
    "violet")
```

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
```

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

```
## [1] "Average profit user get in 1 round per game"
```

```r
print(mean(simple_games[simple_games$Type == 0, ]$CurrGameProfit))
```

```
## [1] -0.7253333
```

```r
print(mean(id_games[id_games$Type == 0, ]$CurrGameProfit))
```

```
## [1] 1.877333
```

```r
print(mean(score_games[score_games$Type == 0, ]$CurrGameProfit))
```

```
## [1] 2.221333
```

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

```
## [1] 2.546667
```

```r
print(mean(simple_games[simple_games$Type == 1, ]$CurrGameProfit))
```

```
## [1] 6.730667
```

```r
print(mean(id_games[id_games$Type == 1, ]$CurrGameProfit))
```

```
## [1] 8.730667
```

```r
print(mean(score_games[score_games$Type == 1, ]$CurrGameProfit))
```

```
## [1] 8.354667
```

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

```
## [1] 8.36
```

```r
print("Response time for games")
```

```
## [1] "Response time for games"
```

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

```
## [1] 9.874856
```

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

```
## [1] 12.73053
```

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

```
## [1] 11.03453
```

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

```
## [1] 15.34007
```

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

```
## [1] 10.90826
```

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

```
## [1] 15.04132
```

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

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

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

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

Basic analysis by role

Date: Thu 15-Jun-2017

```
# calculate game metrics
for (type in 0:1) {
    print(paste("Basic game metric of:", Type_names[type +
        1]))
    # calculate average sending amount per game
    print("Mean of sending amount per games")
    print(mean(simple_games[simple_games$Type == type,
        ]$Contribution))
    print(mean(id_games[id_games$Type == type, ]$Contribution))
    print(mean(score_games[score_games$Type == type,
        ]$Contribution))
    print(mean(combine_games[combine_games$Type ==
        type, ]$Contribution))

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

    # calculate by SHOW_TRUST and SHOW_ID
    print(paste("Mean of send proportion no TRUST:",
        mean(c(simple_games[simple_games$Type == type &
            simple_games$my_send_proportional >= 0,
            ]$my_send_proportional, id_games[id_games$Type ==
            type & id_games$my_send_proportional >=
            0, ]$my_send_proportional))))
    print(paste("Mean of send proportion with TRUST:",
        mean(c(score_games[score_games$Type == type &
            score_games$my_send_proportional >= 0,
            ]$my_send_proportional, combine_games[combine_games$Type ==
            type & combine_games$my_send_proportional >=
            0, ]$my_send_proportional))))

    print(paste("Mean of send proportion no ID:", mean(c(simple_games[simple_games$Type ==
        type & simple_games$my_send_proportional >=
        0, ]$my_send_proportional, score_games[score_games$Type ==
        type & score_games$my_send_proportional >=
        0, ]$my_send_proportional))))
    print(paste("Mean of send proportion with ID:",
        mean(c(id_games[id_games$Type == type & id_games$my_send_proportional >=
            0, ]$my_send_proportional, combine_games[combine_games$Type ==
            type & combine_games$my_send_proportional >=
            0, ]$my_send_proportional))))
```

```
    print("Average profit user get in 1 round per game")
    print(mean(simple_games[simple_games$Type == type,
        ]$CurrGameProfit))
    print(mean(id_games[id_games$Type == type, ]$CurrGameProfit))
    print(mean(score_games[score_games$Type == type,
        ]$CurrGameProfit))
    print(mean(combine_games[combine_games$Type ==
        type, ]$CurrGameProfit))
}
```

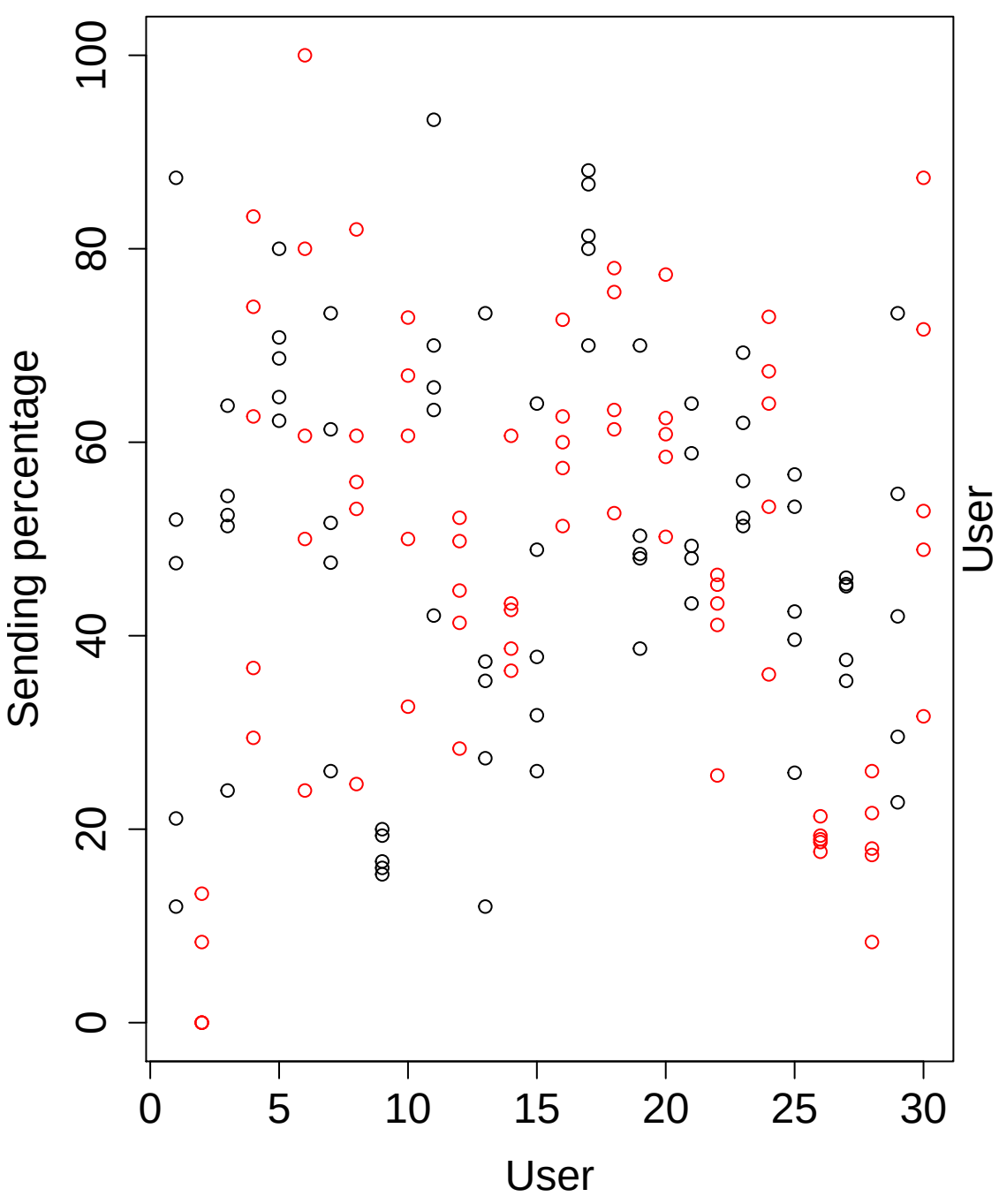
```
## [1] "Basic game metric of: SENDER"
## [1] "Mean of sending amount per games"
## [1] 3.002667
## [1] 5.304
## [1] 5.288
## [1] 5.453333
## [1] "Mean of sending proportion per game"
## [1] 0.3002667
## [1] 0.5304
## [1] 0.5288
## [1] 0.5453333
## [1] "Mean of send proportion no TRUST: 0.415333333333333"
## [1] "Mean of send proportion with TRUST: 0.537066666666667"
## [1] "Mean of send proportion no ID: 0.414533333333333"
## [1] "Mean of send proportion with ID: 0.537866666666667"
## [1] "Average profit user get in 1 round per game"
## [1] -0.7253333
## [1] 1.877333
## [1] 2.221333
## [1] 2.546667
## [1] "Basic game metric of: RECEIVER"
## [1] "Mean of sending amount per games"
## [1] 2.277333
## [1] 7.181333
## [1] 7.509333
## [1] 8
## [1] "Mean of sending proportion per game"
## [1] 0.2615905
## [1] 0.4409586
## [1] 0.476037
## [1] 0.4765017
## [1] "Mean of send proportion no TRUST: 0.36495516097211"
## [1] "Mean of send proportion with TRUST: 0.476271097174619"
## [1] "Mean of send proportion no ID: 0.382636930110798"
## [1] "Mean of send proportion with ID: 0.458437927570962"
## [1] "Average profit user get in 1 round per game"
## [1] 6.730667
## [1] 8.730667
## [1] 8.354667
## [1] 8.36
```

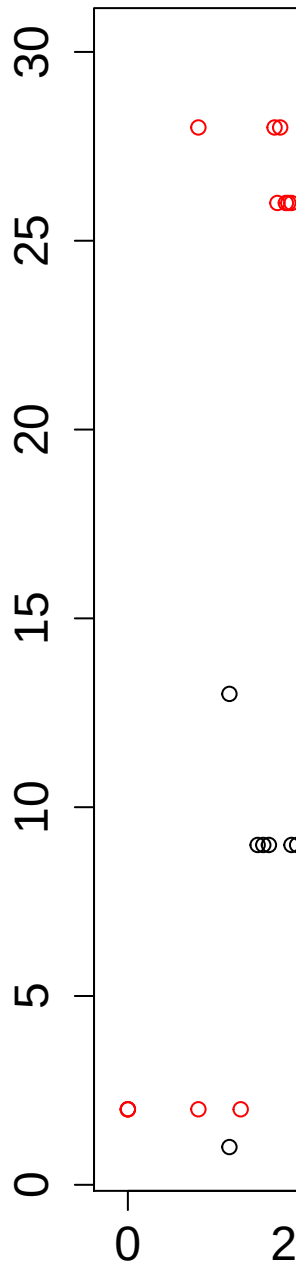
Of course, these above metrics are very basic one. We need to calculate the group interaction on SHOW_TRUST and SHOW_ID variables (both of them are boolean values, mean in a particular game, we show trust score or identity to users or not). In order to do this, we will use 2 - way and 3 - way ANOVA in

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
```



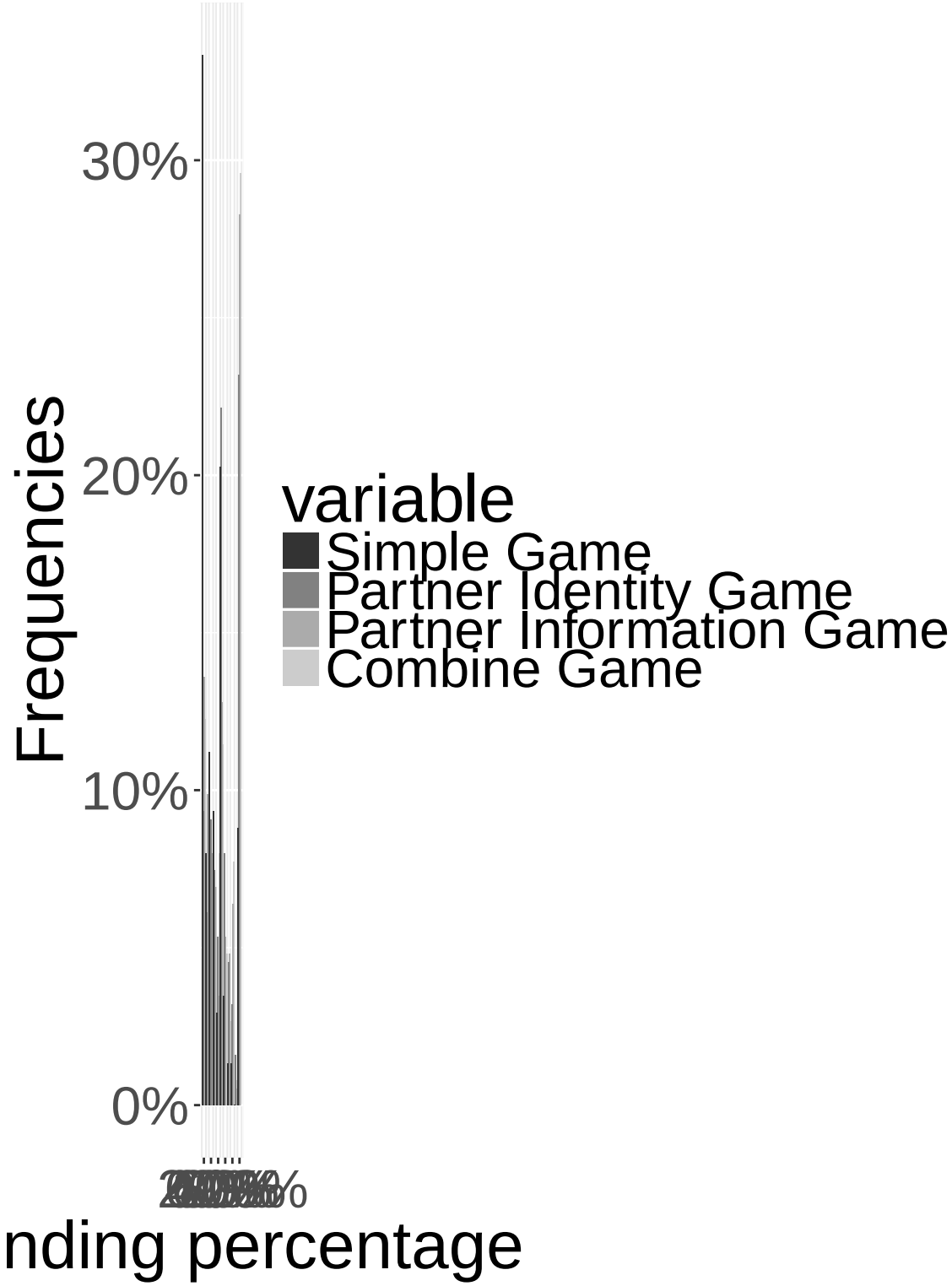


Game metrics for each person of each group

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.

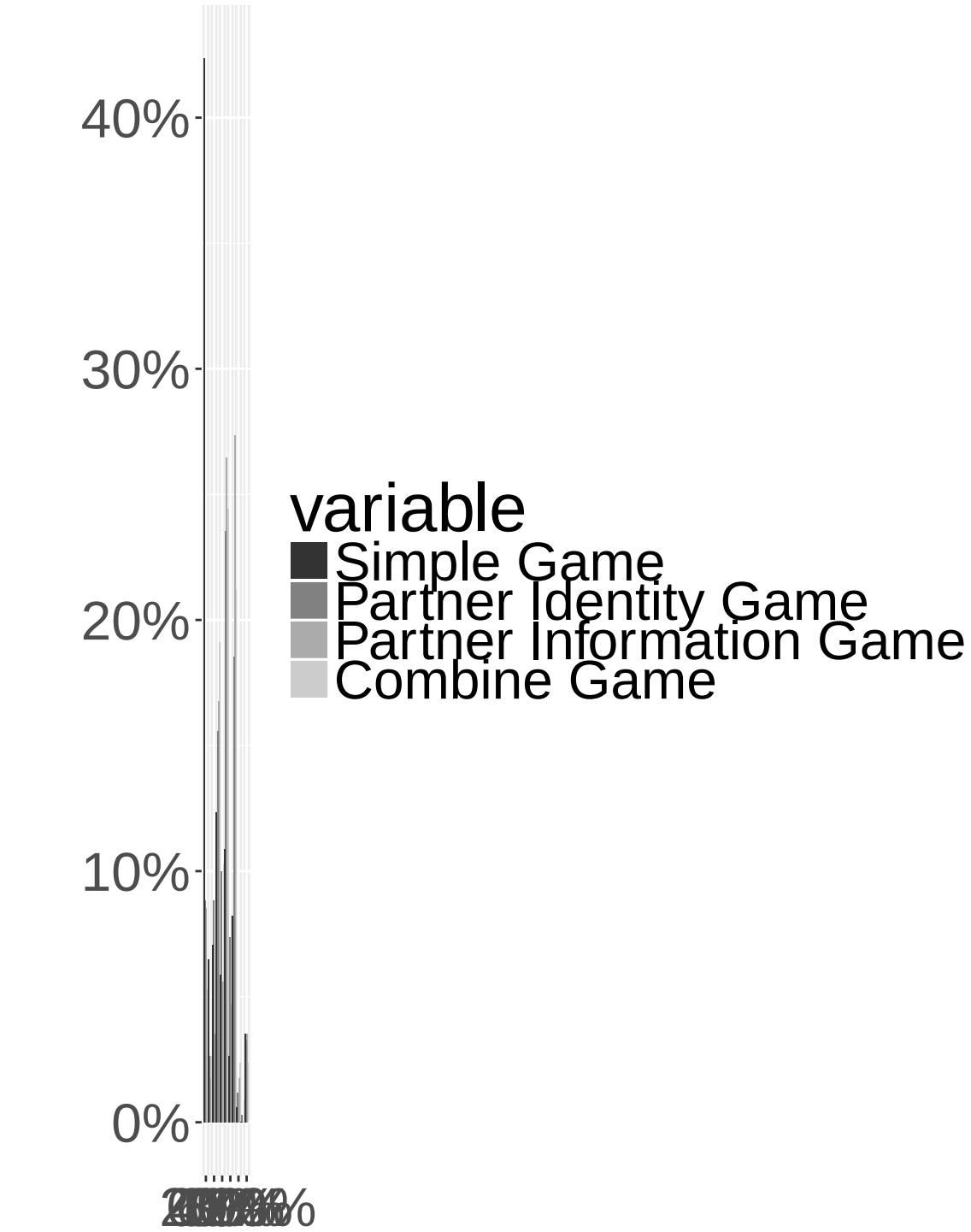
Temporary remove, move to effect of each group.

```
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))
```



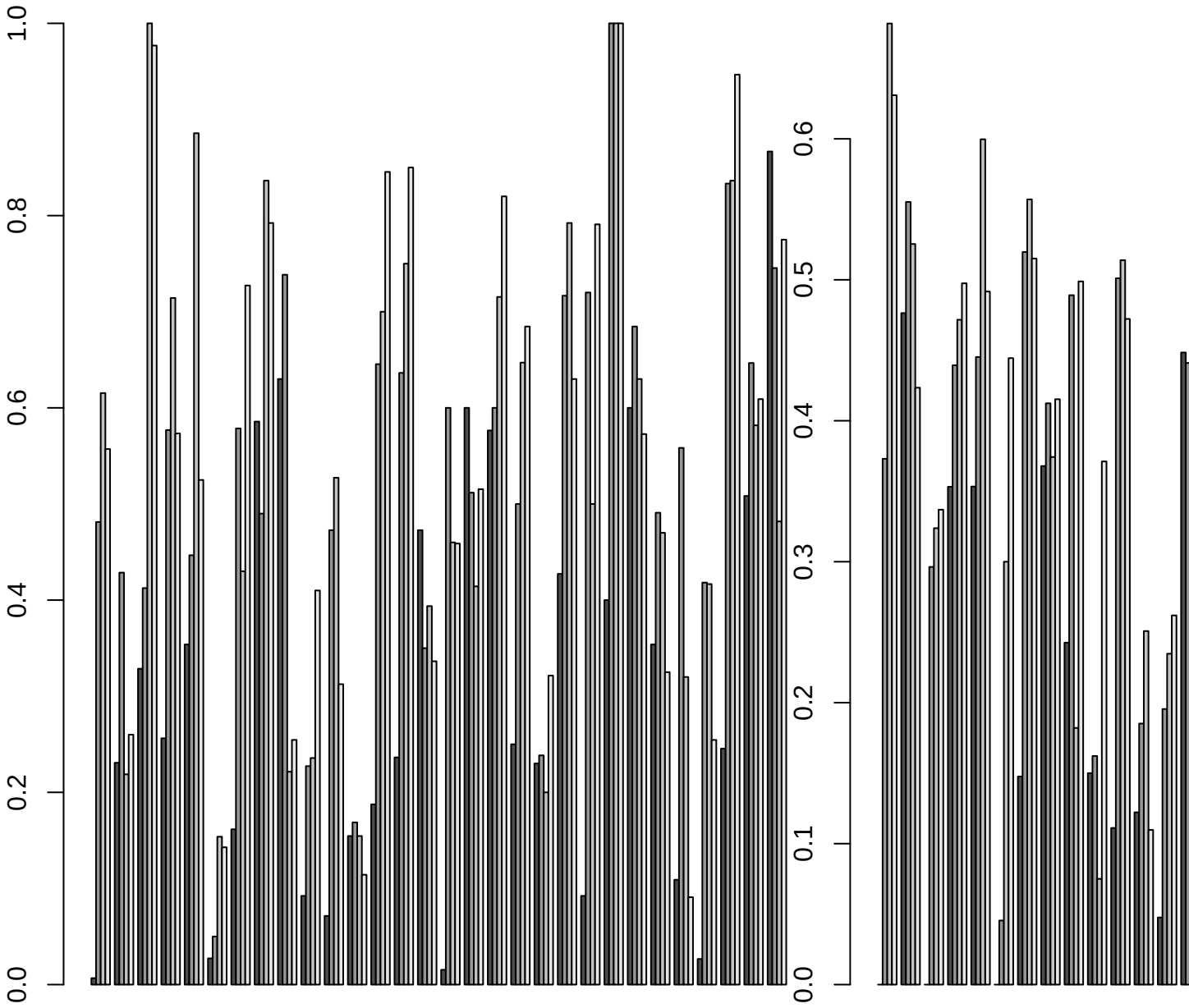

```
# For receivers
c <- cbind(simple_games[simple_games$Type == 1 & simple_games$my_send_proportional >=
    0, ]$my_send_proportional, id_games[id_games$Type ==
    1 & id_games$my_send_proportional >= 0, ]$my_send_proportional,
    score_games[score_games$Type == 1 & score_games$my_send_proportional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




```
for (type in 0:1) {
    sg_senders <- simple_games[simple_games$Type ==
        type & simple_games$my_send_proportional >=
        0, ]
    sg_senders_avg <- aggregate(sg_senders$my_send_proportional,
```

```
        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 homogenity
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-06 ***
## score - id == 0       -0.00160    0.02532  -0.063    1.000
## simple - id == 0      -0.23013    0.02532  -9.088   <1e-06 ***
## simple - score == 0   -0.22853    0.02532  -9.024   <1e-06 ***
## ---
## 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 homogenity
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
```

```r
# 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
```

```r
# 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
```

```r
# 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 homogenity
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 homogenity
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 send proportion on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "Type: SENDER"
## [1] "With GroupID"
##
## Error: id
##           Df Sum Sq Mean Sq F value  Pr(>F)
## GroupID    4  2.004  0.5010   6.189 0.00132 **
## Residuals 25  2.024  0.0809
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST           1 0.4955  0.4955   5.173 0.0853 .
## GroupID:SHOW_TRUST   4 0.3832  0.0958   2.610 0.0597 .
## Residuals           25 0.9177  0.0367
## ---
## 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  16.828 0.0148 *
## GroupID:SHOW_ID   4 0.0982  0.0246   1.263 0.3106
## Residuals        25 0.4858  0.0194
## ---
## 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   8.859 0.0409 *
## GroupID:SHOW_TRUST:SHOW_ID    4 0.1689  0.0422   2.698 0.0538 .
## Residuals                    25 0.3914  0.0157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "------------------------------------------------"
## [1] "------------------------------------------------"
## [1] "ANOVA analysis with send proportion on GroupID:SHOW_TRUST:SHOW_ID"
## [1] "Type: RECEIVER"
## [1] "With GroupID"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4 0.1004 0.02509   0.296  0.878
## Residuals 25 2.1185 0.08474
##
## Error: id:SHOW_TRUST
```

```
##                     Df Sum Sq Mean Sq F value   Pr(>F)
## SHOW_TRUST           1 0.2764 0.27640  74.439 0.000992 ***
## GroupID:SHOW_TRUST   4 0.0149 0.00371   0.153 0.959705
## Residuals           25 0.6053 0.02421
## ---
## 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  35.862 0.00391 **
## GroupID:SHOW_ID   4 0.0273 0.00683   0.553 0.69894
## Residuals        25 0.3088 0.01235
## ---
## 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 0.15135 0.15135   6.966 0.0576 .
## GroupID:SHOW_TRUST:SHOW_ID   4 0.08691 0.02173   2.484 0.0695 .
## Residuals                   25 0.21871 0.00875
## ---
## 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

Now, we create five datasets for five rounds to analyze ANOVA. We expected that the same effect will appear in 5 analysis.

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] "---****---"
```

Test against group effect on each round

Repeat the previous analysis but for all 12 rounds and without group id

Date: Tue 13-Jun-2017

Date: 14-Jun-2017 I added the function anove_random_subject(df) so we can run ANOVA with corrected error term

```
## [1] "Analyze SENDER"
## [1] "Round number: 1"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)         1.1302  1 15.9298 0.0001368 ***
## SHOW_TRUST          0.1215  1  1.7125 0.1941088
## SHOW_ID             0.0042  1  0.0587 0.8090891
## id                  5.7084 29  2.7744 0.0001375 ***
## SHOW_TRUST:SHOW_ID 0.0141  1  0.1985 0.6570421
## Residuals           6.1726 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Mean of all games this round: 0.535833333333333"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 2"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)         1.2445  1 15.6539 0.0001548 ***
## SHOW_TRUST          0.2407  1  3.0273 0.0854091 .
## SHOW_ID             0.1307  1  1.6436 0.2032319
## id                  5.7130 29  2.4780 0.0006359 ***
## SHOW_TRUST:SHOW_ID 0.0480  1  0.6038 0.4392440
## Residuals           6.9163 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Mean of all games this round: 0.49"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 3"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value   Pr(>F)
```

```
## (Intercept)        0.2868  1  3.3086 0.072360 .
## SHOW_TRUST         0.4682  1  5.4010 0.022456 *
## SHOW_ID            0.2535  1  2.9245 0.090808 .
## id                 5.3537 29  2.1298 0.003786 **
## SHOW_TRUST:SHOW_ID 0.1401  1  1.6161 0.207028
## Residuals          7.5412 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.408333333333333"
## [1] "Mean with trust: 0.516666666666667"
## [1] "Mean of all games this round: 0.4625"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 4"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value   Pr(>F)
## (Intercept)        0.1383  1  1.5158 0.221568
## SHOW_TRUST         0.6827  1  7.4839 0.007545 **
## SHOW_ID            0.5607  1  6.1464 0.015096 *
## id                 5.0587 29  1.9123 0.011230 *
## SHOW_TRUST:SHOW_ID 0.2803  1  3.0732 0.083115 .
## Residuals          7.9360 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.43"
## [1] "Mean with trust: 0.546666666666667"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.44"
## [1] "Mean with nickname: 0.536666666666667"
## [1] "Mean of all games this round: 0.488333333333333"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 5"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value   Pr(>F)
## (Intercept)        0.1051  1  1.0906 0.299230
## SHOW_TRUST         0.4507  1  4.6769 0.033315 *
## SHOW_ID            1.0140  1 10.5230 0.001673 **
## id                 6.1620 29  2.2051 0.002583 **
## SHOW_TRUST:SHOW_ID 0.4563  1  4.7357 0.032253 *
```

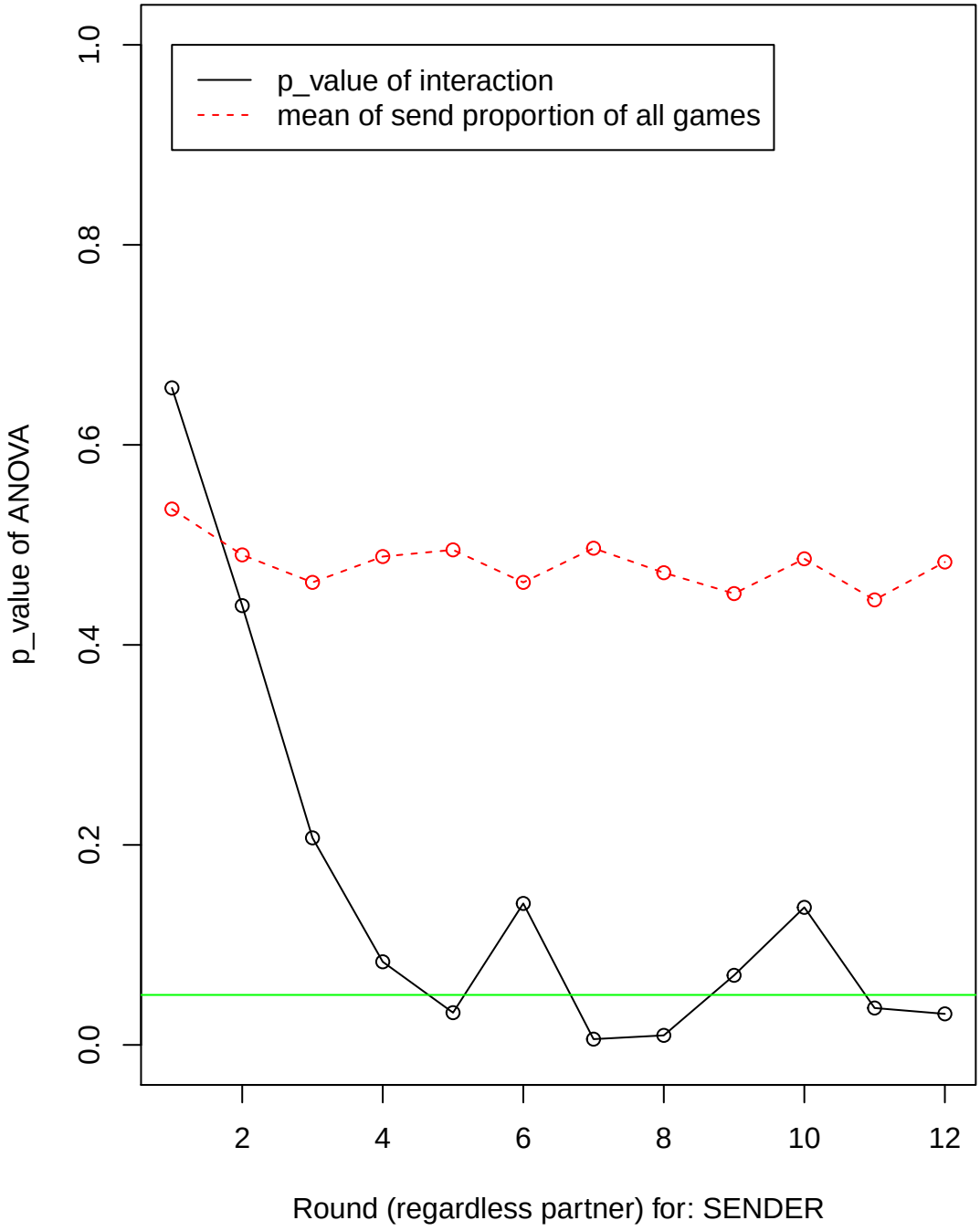
```
## Residuals          8.3833 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.47"
## [1] "Mean with trust: 0.52"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.426666666666667"
## [1] "Mean with nickname: 0.563333333333333"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.34"
## [1] "Mean id 0.6"
## [1] "Mean score 0.513333333333333"
## [1] "Mean combine 0.526666666666667"
## [1] "Mean of all games this round: 0.495"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 6"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value   Pr(>F)
## (Intercept)        0.0591  1  0.6005 0.440495
## SHOW_TRUST         0.4002  1  4.0649 0.046867 *
## SHOW_ID            0.7707  1  7.8285 0.006331 **
## id                 5.6738 29  1.9874 0.007743 **
## SHOW_TRUST:SHOW_ID 0.2168  1  2.2018 0.141466
## Residuals          8.5646 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.423333333333333"
## [1] "Mean with trust: 0.501666666666667"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.391666666666667"
## [1] "Mean with nickname: 0.533333333333333"
## [1] "Mean of all games this round: 0.4625"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 7"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value    Pr(>F)
## (Intercept)        0.3841  1  3.8425 0.0531683 .
## SHOW_TRUST         1.5360  1 15.3665 0.0001761 ***
```

```
## SHOW_ID            1.7682  1 17.6891 6.295e-05 ***
## id                 5.7937 29  1.9987 0.0073203 **
## SHOW_TRUST:SHOW_ID 0.8003  1  8.0067 0.0057857 **
## Residuals          8.6963 87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.418333333333333"
## [1] "Mean with trust: 0.575"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.406666666666667"
## [1] "Mean with nickname: 0.586666666666667"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.246666666666667"
## [1] "Mean id 0.59"
## [1] "Mean score 0.566666666666667"
## [1] "Mean combine 0.583333333333333"
## [1] "Mean of all games this round: 0.496666666666667"
## [1] "Length of each vector"
## [1] 30
## [1] 30
## [1] 30
## [1] 30
## [1] "Analyze SENDER"
## [1] "Round number: 8"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value    Pr(>F)
## (Intercept)        0.0496  1  0.5125 0.4761049
## SHOW_TRUST         1.0082  1 10.4187 0.0017942 **
## SHOW_ID            1.5852  1 16.3810 0.0001167 ***
## id                 6.0613 29  2.1599 0.0035704 **
## SHOW_TRUST:SHOW_ID 0.6816  1  7.0439 0.0095508 **
## Residuals          7.9352 82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.410344827586207"
## [1] "Mean with trust: 0.535087719298246"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.37719298245614"
## [1] "Mean with nickname: 0.56551724137931"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.231034482758621"
## [1] "Mean id 0.589655172413793"
## [1] "Mean score 0.528571428571429"
## [1] "Mean combine 0.541379310344828"
## [1] "Mean of all games this round: 0.472173913043478"
## [1] "Length of each vector"
## [1] 29
## [1] 29
## [1] 28
## [1] 29
```

```
## [1] "Analyze SENDER"
## [1] "Round number: 9"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value   Pr(>F)
## (Intercept)         0.0894  1  0.7949 0.375242
## SHOW_TRUST          1.1014  1  9.7940 0.002425 **
## SHOW_ID             0.4838  1  4.3020 0.041205 *
## id                  6.3387 29  1.9436 0.010311 *
## SHOW_TRUST:SHOW_ID 0.3801  1  3.3801 0.069608 .
## Residuals           9.2217 82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.372413793103448"
## [1] "Mean with trust: 0.531578947368421"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.408771929824561"
## [1] "Mean with nickname: 0.493103448275862"
## [1] "Mean of all games this round: 0.451304347826087"
## [1] "Length of each vector"
## [1] 29
## [1] 29
## [1] 28
## [1] 29
## [1] "Analyze SENDER"
## [1] "Round number: 10"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value   Pr(>F)
## (Intercept)         0.0724  1  0.6453 0.424106
## SHOW_TRUST          0.6784  1  6.0467 0.016037 *
## SHOW_ID             1.2701  1 11.3205 0.001168 **
## id                  6.5414 29  2.0105 0.007447 **
## SHOW_TRUST:SHOW_ID 0.2523  1  2.2490 0.137541
## Residuals           9.1998 82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.41551724137931"
## [1] "Mean with trust: 0.557894736842105"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.380701754385965"
## [1] "Mean with nickname: 0.589655172413793"
## [1] "Mean of all games this round: 0.486086956521739"
## [1] "Length of each vector"
## [1] 29
## [1] 29
## [1] 28
## [1] 29
## [1] "Analyze SENDER"
## [1] "Round number: 11"
```

```
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)         0.2819  1  2.8701 0.0948839 .
## SHOW_TRUST          1.2015  1 12.2317 0.0008392 ***
## SHOW_ID             0.5429  1  5.5269 0.0216770 *
## id                  5.6864 29  1.9963 0.0104071 *
## SHOW_TRUST:SHOW_ID 0.4454  1  4.5341 0.0369032 *
## Residuals           6.5811 67
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.348"
## [1] "Mean with trust: 0.542"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.395918367346939"
## [1] "Mean with nickname: 0.492156862745098"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.224"
## [1] "Mean id 0.472"
## [1] "Mean score 0.575"
## [1] "Mean combine 0.511538461538461"
## [1] "Mean of all games this round: 0.445"
## [1] "Length of each vector"
## [1] 25
## [1] 25
## [1] 24
## [1] 26
## [1] "Analyze SENDER"
## [1] "Round number: 12"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)         0.0905  1  1.8149  0.186116
## SHOW_TRUST          0.4918  1  9.8664  0.003305 **
## SHOW_ID             0.9992  1 20.0476 6.993e-05 ***
## id                  5.7014 29  3.9444 5.715e-05 ***
## SHOW_TRUST:SHOW_ID 0.2505  1  5.0256  0.031057 *
## Residuals           1.8442 37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.438888888888889"
## [1] "Mean with trust: 0.529411764705882"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.34"
## [1] "Mean with nickname: 0.625714285714286"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.238888888888889"
## [1] "Mean id 0.638888888888889"
## [1] "Mean score 0.447058823529412"
## [1] "Mean combine 0.611764705882353"
```

```
## [1] "Mean of all games this round: 0.482857142857143"
## [1] "Length of each vector"
## [1] 18
## [1] 18
## [1] 17
## [1] 17
## [1] "------"
## [1] "TYPE:  SENDER"
```




```
## [1] "Analyze RECEIVER"
## [1] "Round number: 1"
## Anova Table (Type III tests)
```

```
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)        0.20358  1  5.5877   0.02064 *
## SHOW_TRUST         0.13457  1  3.6935   0.05837 .
## SHOW_ID            0.09614  1  2.6389   0.10842
## id                 3.12770 29  2.9603 8.406e-05 ***
## SHOW_TRUST:SHOW_ID 0.02688  1  0.7377   0.39308
## Residuals          2.76892 76
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Mean of all games this round: 0.4295458958303"
## [1] "Length of each vector"
## [1] 26
## [1] 28
## [1] 27
## [1] 28
## [1] "Analyze RECEIVER"
## [1] "Round number: 2"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)        0.45260  1 11.2884 0.0012288 **
## SHOW_TRUST         0.68777  1 17.1540 8.946e-05 ***
## SHOW_ID            0.51637  1 12.8789 0.0005899 ***
## id                 2.02717 29  1.7435 0.0287615 *
## SHOW_TRUST:SHOW_ID 0.29902  1  7.4579 0.0078688 **
## Residuals          3.00706 75
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.356074673055805"
## [1] "Mean with trust: 0.48540404040404"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.373185941043084"
## [1] "Mean with nickname: 0.462424894628284"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.231642512077295"
## [1] "Mean id 0.45147266313933"
## [1] "Mean score 0.498397435897436"
## [1] "Mean combine 0.473754789272031"
## [1] "Mean of all games this round: 0.421936850872036"
## [1] "Length of each vector"
## [1] 23
## [1] 30
## [1] 26
## [1] 29
## [1] "Analyze RECEIVER"
## [1] "Round number: 3"
## Anova Table (Type III tests)
##
## Response: score
##                    Sum Sq Df F value   Pr(>F)
```

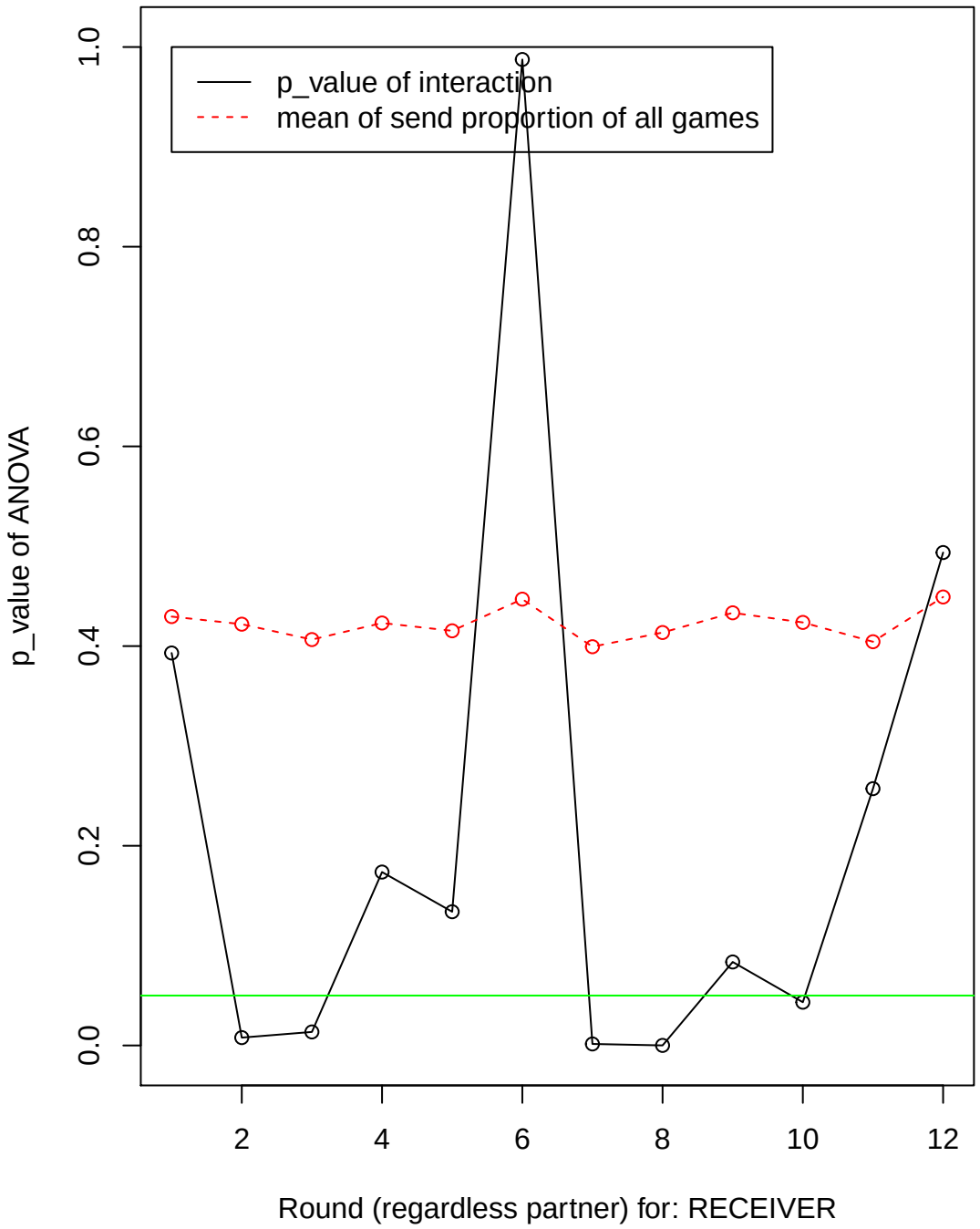
```
## (Intercept)        0.0638  1  1.2174 0.273396
## SHOW_TRUST         0.5067  1  9.6717 0.002645 **
## SHOW_ID            0.6661  1 12.7142 0.000636 ***
## id                 3.2181 29  2.1182 0.005030 **
## SHOW_TRUST:SHOW_ID 0.3350  1  6.3952 0.013544 *
## Residuals          3.9291 75
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.3684561245882"
## [1] "Mean with trust: 0.443181818181818"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.343194070080863"
## [1] "Mean with nickname: 0.467525252525253"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.223238095238095"
## [1] "Mean id 0.498115079365079"
## [1] "Mean score 0.450297619047619"
## [1] "Mean combine 0.435802469135802"
## [1] "Mean of all games this round: 0.40651087595532"
## [1] "Length of each vector"
## [1] 25
## [1] 28
## [1] 28
## [1] 27
## [1] "Analyze RECEIVER"
## [1] "Round number: 4"
## Anova Table (Type III tests)
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)        0.35548  1  8.9165 0.0038759 **
## SHOW_TRUST         0.33863  1  8.4939 0.0047632 **
## SHOW_ID            0.23236  1  5.8284 0.0183496 *
## id                 2.88708 29  2.4971 0.0009286 ***
## SHOW_TRUST:SHOW_ID 0.07530  1  1.8886 0.1736772
## Residuals          2.83058 71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.355646494708995"
## [1] "Mean with trust: 0.480952380952381"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.37983789260385"
## [1] "Mean with nickname: 0.458806739069897"
## [1] "Mean of all games this round: 0.423118894993895"
## [1] "Length of each vector"
## [1] 20
## [1] 28
## [1] 27
## [1] 29
## [1] "Analyze RECEIVER"
## [1] "Round number: 5"
## Anova Table (Type III tests)
```

```
##
## Response: score
##                     Sum Sq Df F value    Pr(>F)
## (Intercept)         0.5005  1 12.5195 0.0007626 ***
## SHOW_TRUST          0.1330  1  3.3275 0.0728752 .
## SHOW_ID             0.2834  1  7.0877 0.0098379 **
## id                  3.2169 29  2.7747 0.0003645 ***
## SHOW_TRUST:SHOW_ID 0.0922  1  2.3052 0.1339407
## Residuals           2.5186 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.377310405643739"
## [1] "Mean with nickname: 0.448832866479925"
## [1] "Mean of all games this round: 0.415306712962963"
## [1] "Length of each vector"
## [1] 21
## [1] 27
## [1] 24
## [1] 24
## [1] "Analyze RECEIVER"
## [1] "Round number: 6"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)         0.20398  1  6.2895 0.0147344 *
## SHOW_TRUST          0.02463  1  0.7595 0.3867863
## SHOW_ID             0.19409  1  5.9847 0.0172335 *
## id                  2.78973 29  2.9662 0.0001581 ***
## SHOW_TRUST:SHOW_ID 0.00001  1  0.0002 0.9874919
## Residuals           2.04319 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.366056910569106"
## [1] "Mean with nickname: 0.507361712361712"
## [1] "Mean of all games this round: 0.44701278659612"
## [1] "Length of each vector"
## [1] 18
## [1] 29
## [1] 23
## [1] 26
## [1] "Analyze RECEIVER"
## [1] "Round number: 7"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)         0.01909  1  0.4538  0.503001
## SHOW_TRUST          0.78458  1 18.6555 5.657e-05 ***
## SHOW_ID             1.10821  1 26.3507 2.955e-06 ***
## id                  1.70033 29  1.3941  0.135741
## SHOW_TRUST:SHOW_ID 0.46311  1 11.0115  0.001508 **
```

```
## Residuals          2.64955 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.334844209288654"
## [1] "Mean with trust: 0.45629474011827"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.326925025329281"
## [1] "Mean with nickname: 0.468847856602959"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.175661375661376"
## [1] "Mean id 0.474129188712522"
## [1] "Mean score 0.449099511599512"
## [1] "Mean combine 0.463777777777778"
## [1] "Mean of all games this round: 0.399364803791887"
## [1] "Length of each vector"
## [1] 21
## [1] 24
## [1] 26
## [1] 25
## [1] "Analyze RECEIVER"
## [1] "Round number: 8"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)        0.07906  1  2.7057   0.10514
## SHOW_TRUST         0.73593  1 25.1842 4.801e-06 ***
## SHOW_ID            0.69912  1 23.9246 7.647e-06 ***
## id                 1.31498 27  1.6667   0.05049 .
## SHOW_TRUST:SHOW_ID 0.54178  1 18.5403 6.138e-05 ***
## Residuals          1.78253 61
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.358292181069959"
## [1] "Mean with trust: 0.466523134076326"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.359259259259259"
## [1] "Mean with nickname: 0.453558949785365"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.188888888888889"
## [1] "Mean id 0.471227709190672"
## [1] "Mean score 0.505291005291005"
## [1] "Mean combine 0.435210622710623"
## [1] "Mean of all games this round: 0.413584080975385"
## [1] "Length of each vector"
## [1] 18
## [1] 27
## [1] 21
## [1] 26
## [1] "Analyze RECEIVER"
## [1] "Round number: 9"
## Anova Table (Type III tests)
```

```
##
## Response: score
##                      Sum Sq Df F value   Pr(>F)
## (Intercept)        0.37230  1  9.9450 0.002573 **
## SHOW_TRUST         0.31236  1  8.3438 0.005462 **
## SHOW_ID            0.48486  1 12.9515 0.000670 ***
## id                 2.43105 29  2.2392 0.004686 **
## SHOW_TRUST:SHOW_ID 0.11607  1  3.1004 0.083636 .
## Residuals          2.13388 57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.365130023640662"
## [1] "Mean with trust: 0.507890365448505"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.376303854875283"
## [1] "Mean with nickname: 0.483242394179894"
## [1] "Mean of all games this round: 0.433337742504409"
## [1] "Length of each vector"
## [1] 20
## [1] 27
## [1] 22
## [1] 21
## [1] "Analyze RECEIVER"
## [1] "Round number: 10"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)        0.47142  1 15.3192 0.0002524 ***
## SHOW_TRUST         0.58890  1 19.1369 5.472e-05 ***
## SHOW_ID            0.21277  1  6.9141 0.0110673 *
## id                 1.96321 28  2.2784 0.0045478 **
## SHOW_TRUST:SHOW_ID 0.13163  1  4.2773 0.0433397 *
## Residuals          1.69252 55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.338472222222222"
## [1] "Mean with trust: 0.496225937183384"
## [1] "SHOW ID significant"
## [1] "Mean no nickname: 0.418462301587302"
## [1] "Mean with nickname: 0.428149273893955"
## [1] "SHOW_TRUST*SHOW_ID significant"
## [1] "Mean simple 0.270261437908497"
## [1] "Mean id 0.388888888888889"
## [1] "Mean score 0.528002070393375"
## [1] "Mean combine 0.46577380952381"
## [1] "Mean of all games this round: 0.42369549352308"
## [1] "Length of each vector"
## [1] 17
## [1] 23
## [1] 23
## [1] 24
```

```
## [1] "Analyze RECEIVER"
## [1] "Round number: 11"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value   Pr(>F)
## (Intercept)        0.16715  1  4.0557 0.050458 .
## SHOW_TRUST         0.30568  1  7.4169 0.009372 **
## SHOW_ID            0.06416  1  1.5568 0.219041
## id                 2.24642 29  1.8795 0.030301 *
## SHOW_TRUST:SHOW_ID 0.05438  1  1.3195 0.257188
## Residuals          1.73100 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SHOW TRUST significant"
## [1] "Mean no trust: 0.320138888888889"
## [1] "Mean with trust: 0.46702657807309"
## [1] "Mean of all games this round: 0.404354497354497"
## [1] "Length of each vector"
## [1] 12
## [1] 20
## [1] 23
## [1] 20
## [1] "Analyze RECEIVER"
## [1] "Round number: 12"
## Anova Table (Type III tests)
##
## Response: score
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)        1.12578  1 27.6618 1.021e-05 ***
## SHOW_TRUST         0.01109  1  0.2725   0.60538
## SHOW_ID            0.04150  1  1.0197   0.32040
## id                 2.28733 29  1.9380   0.03668 *
## SHOW_TRUST:SHOW_ID 0.01953  1  0.4798   0.49369
## Residuals          1.26164 31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Mean of all games this round: 0.449175347222222"
## [1] "Length of each vector"
## [1] 9
## [1] 18
## [1] 17
## [1] 20
## [1] "------"
## [1] "TYPE:  RECEIVER"
```


p_value of interaction
mean of send proportion of all games
p_value of ANOVA
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
12
Round (regardless partner) for: RECEIVER

ANOVA on each round to check the group effect

Date: 14-Jun-2017 It is only available for SENDER because for RECEIVER the data is imbalanced (due to zero transaction removal)

```
## [1] "Analyze SENDER"
## [1] "Round number: 1"
##
## Error: id
##           Df Sum Sq Mean Sq F value  Pr(>F)
## GroupID    4  2.421  0.6051   4.601 0.00637 **
## Residuals 25  3.288  0.1315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST           1 0.1401 0.14008   1.270  0.323
## GroupID:SHOW_TRUST   4 0.4412 0.11029   1.861  0.149
## Residuals           25 1.4812 0.05925
##
## Error: id:SHOW_ID
##                  Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID           1 0.0008 0.00075   0.009  0.930
## GroupID:SHOW_ID   4 0.3472 0.08679   0.925  0.465
## Residuals        25 2.3446 0.09378
##
## Error: id:SHOW_TRUST:SHOW_ID
##                              Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID            1 0.0141 0.01408   0.167  0.704
## GroupID:SHOW_TRUST:SHOW_ID    4 0.3372 0.08429   1.726  0.176
## Residuals                    25 1.2212 0.04885
## [1] "Analyze SENDER"
## [1] "Round number: 2"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4  1.419  0.3547   2.065  0.116
## Residuals 25  4.294  0.1718
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST           1 0.2253 0.22533   0.785 0.4258
## GroupID:SHOW_TRUST   4 1.1488 0.28721   3.536 0.0203 *
## Residuals           25 2.0308 0.08123
## ---
## 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 0.0853 0.08533   2.022  0.228
## GroupID:SHOW_ID   4 0.1688 0.04221   0.713  0.591
## Residuals        25 1.4808 0.05923
##
```

```
## Error: id:SHOW_TRUST:SHOW_ID
##                            Df Sum Sq Mean Sq F value  Pr(>F)
## SHOW_TRUST:SHOW_ID          1 0.0480 0.04800   0.219 0.66407
## GroupID:SHOW_TRUST:SHOW_ID  4 0.8762 0.21904   4.523 0.00692 **
## Residuals                  25 1.2108 0.04843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze SENDER"
## [1] "Round number: 3"
##
## Error: id
##           Df Sum Sq Mean Sq F value  Pr(>F)
## GroupID    4  2.553  0.6381   5.695 0.00212 **
## Residuals 25  2.801  0.1120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST           1 0.3521  0.3521   5.469 0.0795 .
## GroupID:SHOW_TRUST   4 0.2575  0.0644   0.755 0.5645
## Residuals           25 2.1329  0.0853
## ---
## 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 0.1141 0.11408   1.924  0.238
## GroupID:SHOW_ID   4 0.2372 0.05929   0.572  0.685
## Residuals        25 2.5912 0.10365
##
## Error: id:SHOW_TRUST:SHOW_ID
##                            Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST:SHOW_ID          1 0.1401 0.14008   1.336  0.312
## GroupID:SHOW_TRUST:SHOW_ID  4 0.4195 0.10488   1.378  0.270
## Residuals                  25 1.9029 0.07612
## [1] "Analyze SENDER"
## [1] "Round number: 4"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4  1.333  0.3332   2.236 0.0939 .
## Residuals 25  3.726  0.1490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST           1 0.4083  0.4083   3.197  0.148
## GroupID:SHOW_TRUST   4 0.5108  0.1277   1.327  0.287
## Residuals           25 2.4058  0.0962
##
## Error: id:SHOW_ID
##                  Df Sum Sq Mean Sq F value Pr(>F)
```

```
## SHOW_ID          1 0.2803 0.28033   5.327 0.0822 .
## GroupID:SHOW_ID  4 0.2105 0.05263   0.964 0.4443
## Residuals       25 1.3642 0.05457
## ---
## 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 0.2803 0.28033   1.128 0.3480
## GroupID:SHOW_TRUST:SHOW_ID  4 0.9938 0.24846   2.534 0.0654 .
## Residuals                  25 2.4508 0.09803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze SENDER"
## [1] "Round number: 5"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4  1.274  0.3184   1.628  0.198
## Residuals 25  4.888  0.1955
##
## Error: id:SHOW_TRUST
##                    Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST          1  0.075 0.07500   0.984  0.377
## GroupID:SHOW_TRUST  4  0.305 0.07625   0.533  0.713
## Residuals          25  3.575 0.14300
##
## Error: id:SHOW_ID
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## SHOW_ID          1 0.5603  0.5603 480.286 2.57e-05 ***
## GroupID:SHOW_ID  4 0.0047  0.0012   0.016    0.999
## Residuals       25 1.8500  0.0740
## ---
## 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 0.4563  0.4563   2.354 0.1997
## GroupID:SHOW_TRUST:SHOW_ID  4 0.7753  0.1938   2.587 0.0614 .
## Residuals                  25 1.8733  0.0749
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze SENDER"
## [1] "Round number: 6"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4  2.152  0.5379   3.818 0.0148 *
## Residuals 25  3.522  0.1409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                    Df Sum Sq Mean Sq F value Pr(>F)
```

```
## SHOW_TRUST          1  0.184  0.1841   1.211  0.333
## GroupID:SHOW_TRUST  4  0.608  0.1520   1.180  0.344
## Residuals          25  3.220  0.1288
##
## Error: id:SHOW_ID
##                 Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID          1 0.6021  0.6021   8.865 0.0408 *
## GroupID:SHOW_ID  4 0.2717  0.0679   0.739 0.5745
## Residuals       25 2.2988  0.0920
## ---
## 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 0.2167 0.21675   7.306 0.0539 .
## GroupID:SHOW_TRUST:SHOW_ID  4 0.1187 0.02967   0.362 0.8330
## Residuals                  25 2.0471 0.08188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Analyze SENDER"
## [1] "Round number: 7"
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## GroupID    4  2.039  0.5097   3.393 0.0239 *
## Residuals 25  3.755  0.1502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: id:SHOW_TRUST
##                     Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_TRUST          1  0.736  0.7363   3.519  0.134
## GroupID:SHOW_TRUST  4  0.837  0.2093   1.639  0.196
## Residuals          25  3.192  0.1277
##
## Error: id:SHOW_ID
##                 Df Sum Sq Mean Sq F value Pr(>F)
## SHOW_ID          1 0.9720  0.9720   5.879 0.0724 .
## GroupID:SHOW_ID  4 0.6613  0.1653   2.005 0.1246
## Residuals       25 2.0617  0.0825
## ---
## 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 0.8003  0.8003  18.868 0.0122 *
## GroupID:SHOW_TRUST:SHOW_ID  4 0.1697  0.0424   0.597 0.6679
## Residuals                  25 1.7750  0.0710
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 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_trust_value + trust_value, data = x_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 ***
## my_trust_value   1.4889     0.1163  12.801 5.57e-13 ***
## trust_value      0.2562     0.1554   1.649 0.110772
## ---
## 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 reputation value of ID Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value, data = x_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 *
## my_trust_value   1.4517     0.1559   9.311 6.41e-10 ***
## trust_value      0.5488     0.3173   1.730   0.0951 .
## ---
## 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 reputation value of Score Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value, data = x_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 ***
## my_trust_value   1.3565     0.1843   7.360 6.45e-08 ***
## trust_value      1.3192     0.2320   5.686 4.85e-06 ***
## ---
## 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 reputation value of Combine Game for type:  SENDER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value, data = x_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 ***
## my_trust_value   1.4954     0.1795   8.332 6.10e-09 ***
## trust_value      1.0693     0.2279   4.692 6.96e-05 ***
## ---
## 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 reputation value of Simple Game for type:  RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + AbsPartnerSend,
##     data = x_simple)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21443 -0.10224 -0.01646  0.06848  0.35236
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.02870    0.14768   0.194   0.8474
## my_trust_value  1.16611    0.19427   6.003 2.45e-06 ***
## trust_value     0.34388    0.50019   0.687   0.4979
## AbsPartnerSend -0.06616    0.02989  -2.214   0.0358 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1436 on 26 degrees of freedom
## Multiple R-squared:  0.6096, Adjusted R-squared:  0.5646
## F-statistic: 13.53 on 3 and 26 DF,  p-value: 1.635e-05
##
## [1] "Linear regression of relative sending on reputation value of ID Game for type:  RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + AbsPartnerSend,
##     data = x_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18551 -0.04782 -0.01754  0.05094  0.15162
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.19837    0.17042  -1.164   0.2550
## my_trust_value  1.34544    0.15056   8.936 2.09e-09 ***
## trust_value     0.41643    0.42598   0.978   0.3373
## AbsPartnerSend -0.02995    0.01620  -1.849   0.0759 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0842 on 26 degrees of freedom
## Multiple R-squared:  0.7721, Adjusted R-squared:  0.7458
## F-statistic: 29.36 on 3 and 26 DF,  p-value: 1.662e-08
##
## [1] "Linear regression of relative sending on reputation value of Score Game for type:  RECEIVER"
##
## Call:
## lm(formula = RelSend ~ my_trust_value + trust_value + AbsPartnerSend,
##     data = x_score)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.229257 -0.063228 -0.004027  0.079191  0.242774
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.03859    0.23987  -0.161    0.873
## my_trust_value  1.36838    0.29637   4.617 9.23e-05 ***
## trust_value     0.16777    0.70654   0.237    0.814
## AbsPartnerSend -0.04135    0.02815  -1.469    0.154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1263 on 26 degrees of freedom
## Multiple R-squared:  0.4754, Adjusted R-squared:  0.4149
## F-statistic: 7.854 on 3 and 26 DF,  p-value: 0.0006825
##
## [1] "Linear regression of relative sending on reputation value of Combine Game for type:  RECEIVER"
##
## Call:
```

```
## lm(formula = RelSend ~ my_trust_value + trust_value + AbsPartnerSend,
##     data = x_combine)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.177927 -0.064764 -0.005558  0.069571  0.136813
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.36764    0.12970   2.835 0.008761 **
## my_trust_value  0.81176    0.20674   3.927 0.000566 ***
## trust_value    -0.77274    0.35815  -2.158 0.040378 *
## AbsPartnerSend  0.01111    0.01892   0.587 0.562176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09411 on 26 degrees of freedom
## Multiple R-squared:  0.5463, Adjusted R-squared:  0.494
## F-statistic: 10.44 on 3 and 26 DF,  p-value: 0.0001097
```

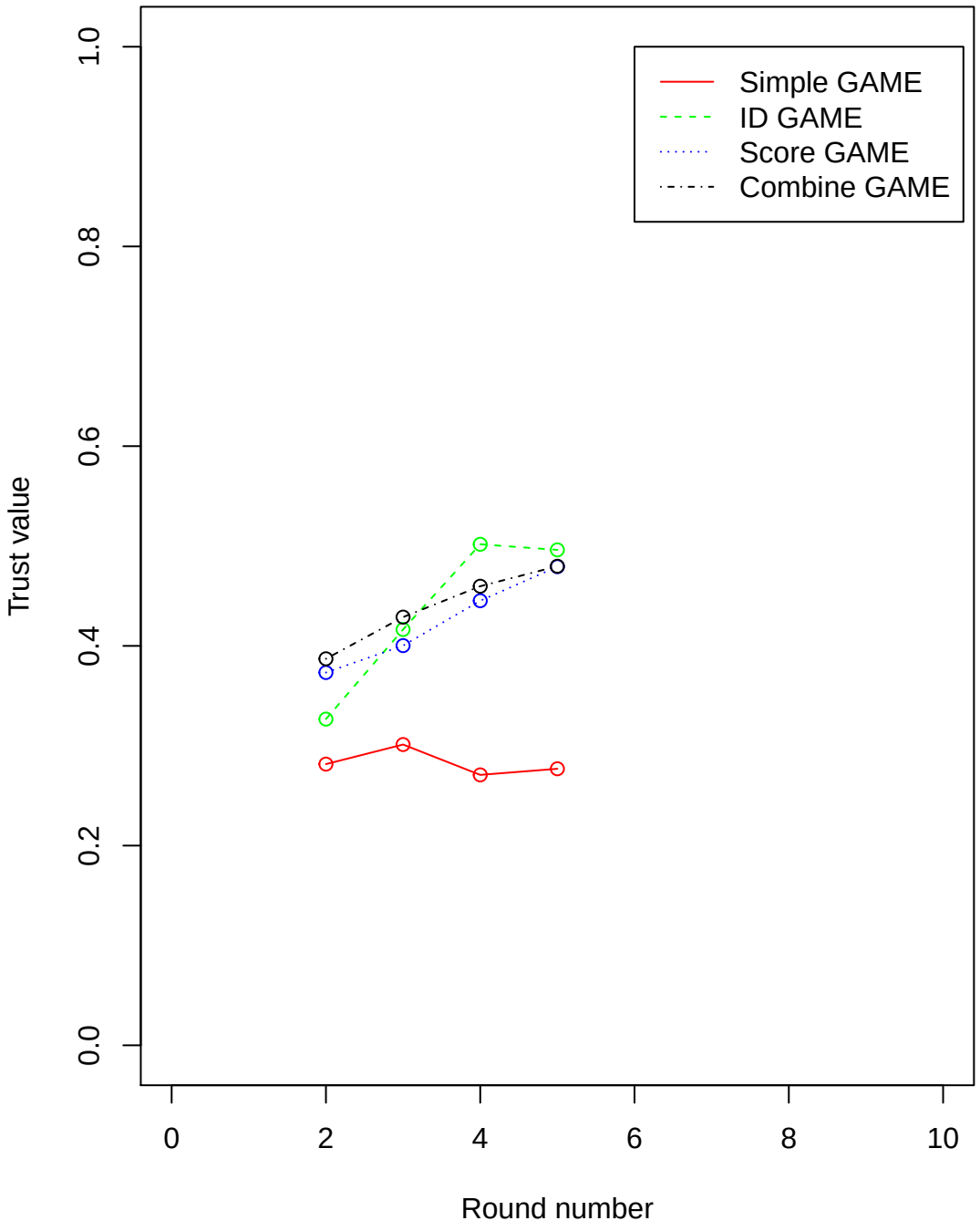
Analyze by individual trial rather than average value

Trust score over time & Chi-square on frequency of sending proportion by game

Date: Tue 13-Jun-2017 Test the amound sent based on trust value and amount received (if any) using individual but not average sending proportion

Update on Thu 15-Jun-2017

```
## [1] "Number of sending 0.0 as: SENDER"
## [1] "Simple GAME : 125"
## [1] "ID GAME : 35"
## [1] "Score GAME : 51"
## [1] "Combine GAME : 46"
##      [,1] [,2]
## [1,]  125   35
## [2,]   51   46
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 17.097, df = 1, p-value = 3.552e-05
##
## [1] "Number of sending >= 0.7 as: SENDER"
## [1] "Simple GAME : 43"
## [1] "ID GAME : 122"
## [1] "Score GAME : 150"
## [1] "Combine GAME : 153"
##      [,1] [,2]
## [1,]   43  122
## [2,]  150  153
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 23.272, df = 1, p-value = 1.406e-06
##
## [1] "Number of sending 1.0 (maximum) as: SENDER"
## [1] "Simple GAME : 33"
## [1] "ID GAME : 87"
## [1] "Score GAME : 106"
## [1] "Combine GAME : 111"
##      [,1] [,2]
## [1,]   43  122
## [2,]  150  153
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 23.272, df = 1, p-value = 1.406e-06
```

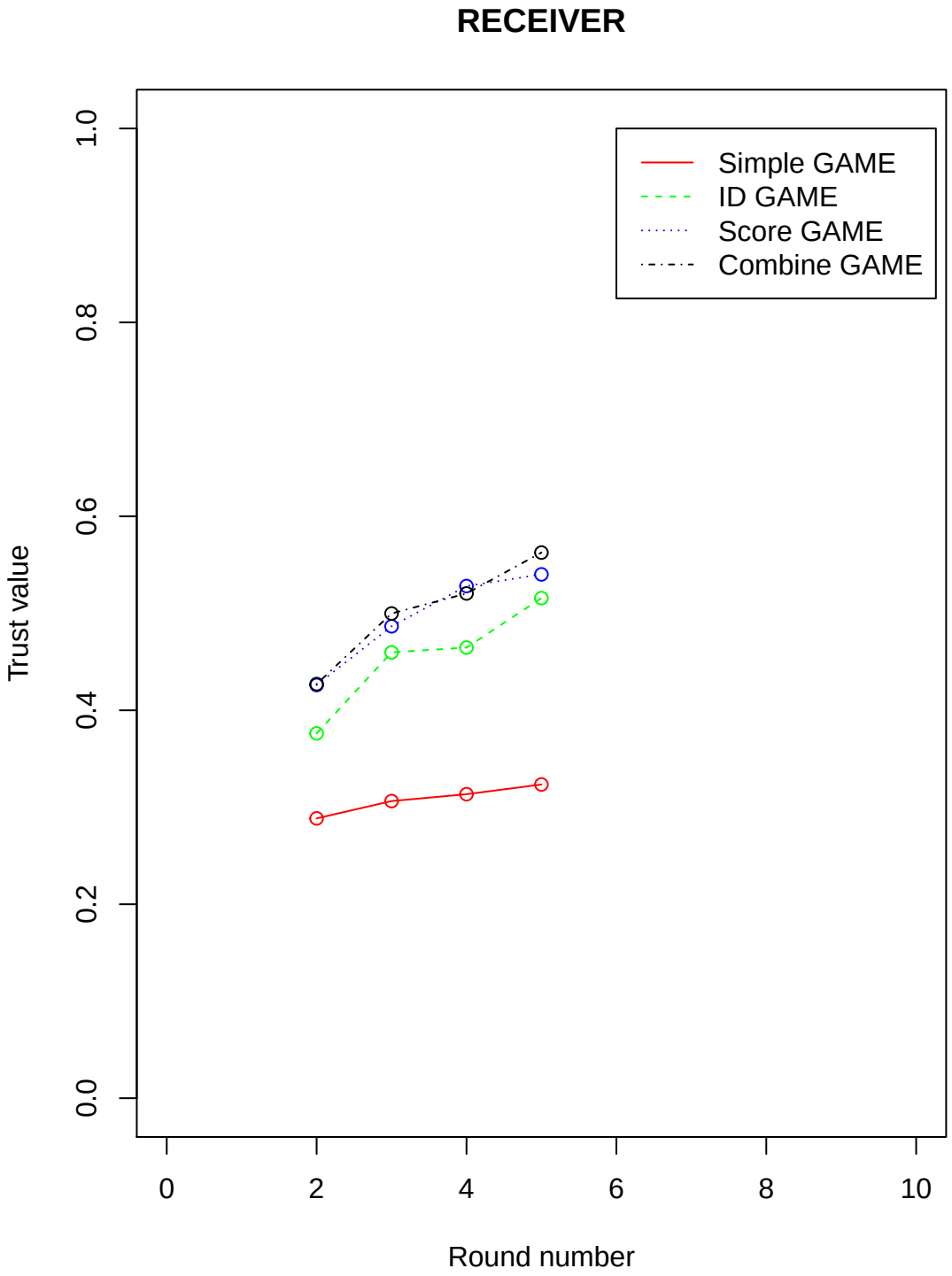



```
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Simple GAME SEN
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43284 -0.09055  0.00045  0.07011  0.49822
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.040044   0.027253   1.469    0.143
## rounds      -0.011441   0.007417  -1.543    0.124
##
## Residual standard error: 0.1436 on 298 degrees of freedom
## Multiple R-squared:  0.007921,   Adjusted R-squared:  0.004592
## F-statistic: 2.379 on 1 and 298 DF,  p-value: 0.124
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: ID GAME SENDER"
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47395 -0.06351  0.00843  0.08878  0.45922
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.177745   0.028348  -6.270 1.26e-09 ***
## rounds       0.050784   0.007715   6.582 2.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1494 on 298 degrees of freedom
## Multiple R-squared:  0.1269, Adjusted R-squared:  0.124
## F-statistic: 43.33 on 1 and 298 DF,  p-value: 2.086e-10
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Score GAME SENDE
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4783 -0.1162  0.0161  0.1264  0.5313
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.121773   0.031788  -3.831 0.000156 ***
## rounds       0.034792   0.008652   4.021 7.33e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1675 on 298 degrees of freedom
## Multiple R-squared:  0.05148,    Adjusted R-squared:  0.04829
## F-statistic: 16.17 on 1 and 298 DF,  p-value: 7.334e-05
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Combine GAME SEN
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40674 -0.10245  0.00660  0.08397  0.59546
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.103163   0.028193  -3.659 0.000299 ***
## rounds       0.029475   0.007673   3.841 0.000150 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1486 on 298 degrees of freedom
## Multiple R-squared:  0.04718,    Adjusted R-squared:  0.04398
## F-statistic: 14.76 on 1 and 298 DF,  p-value: 0.0001495
##
## [1] "Number of sending 0.0 as: RECEIVER"
## [1] "Simple GAME : 92"
## [1] "ID GAME : 29"
## [1] "Score GAME : 27"
## [1] "Combine GAME : 15"
##      [,1] [,2]
## [1,]   92   29
## [2,]   27   15
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 1.6278, df = 1, p-value = 0.202
##
## [1] "Number of sending >= 0.7 as: RECEIVER"
## [1] "Simple GAME : 9"
## [1] "ID GAME : 17"
## [1] "Score GAME : 20"
## [1] "Combine GAME : 19"
##      [,1] [,2]
## [1,]    9   17
## [2,]   20   19
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 1.144, df = 1, p-value = 0.2848
##
## [1] "Number of sending 1.0 (maximum) as: RECEIVER"
## [1] "Simple GAME : 7"
## [1] "ID GAME : 11"
## [1] "Score GAME : 11"
## [1] "Combine GAME : 7"
##      [,1] [,2]
## [1,]    9   17
## [2,]   20   19
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  x
## X-squared = 1.144, df = 1, p-value = 0.2848
```




```
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Simple GAME RECI
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39246 -0.07447  0.00322  0.07018  0.55105
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.026730   0.034596  -0.773    0.441
## rounds       0.007837   0.009634   0.813    0.417
##
## Residual standard error: 0.1492 on 188 degrees of freedom
## Multiple R-squared:  0.003508,   Adjusted R-squared:  -0.001793
## F-statistic: 0.6618 on 1 and 188 DF,  p-value: 0.417
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: ID GAME RECEIVE
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44929 -0.06160 -0.00610  0.05829  0.33700
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.152876   0.025569  -5.979 7.23e-09 ***
## rounds       0.044032   0.007007   6.284 1.35e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1285 on 265 degrees of freedom
## Multiple R-squared:  0.1297, Adjusted R-squared:  0.1264
## F-statistic: 39.49 on 1 and 265 DF,  p-value: 1.35e-09
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Score GAME RECE
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44424 -0.08353  0.00415  0.09482  0.38899
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.156932   0.031965   -4.91 1.65e-06 ***
## rounds       0.044485   0.008638    5.15 5.28e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1533 on 250 degrees of freedom
## Multiple R-squared:  0.09592,    Adjusted R-squared:  0.0923
## F-statistic: 26.52 on 1 and 250 DF,  p-value: 5.28e-07
##
## [1] "Regression of resid trust score on round after elimiate subject effect of game: Combine GAME RE
```

```
##
## Call:
## lm(formula = resid ~ rounds, data = ldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49744 -0.06710  0.01748  0.08119  0.53212
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.14789    0.02695  -5.488 9.76e-08 ***
## rounds       0.04244    0.00736   5.767 2.32e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.133 on 256 degrees of freedom
## Multiple R-squared:  0.115,  Adjusted R-squared:  0.1115
## F-statistic: 33.26 on 1 and 256 DF,  p-value: 2.317e-08
```

Predict send proportion based on trust scores

```
## [1] "Analyze:  SENDER for game: Simple GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47986 -0.11407 -0.01979  0.08567  0.88872
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.028390   0.025236  -1.125    0.262
## my_trust_value     -0.009743   0.074661  -0.130    0.896
## my_reputation       0.069183   0.074735   0.926    0.355
## trust_value         0.084659   0.073946   1.145    0.253
## partner_reputation -0.044964   0.077666  -0.579    0.563
##
## Residual standard error: 0.1955 on 295 degrees of freedom
## Multiple R-squared:  0.01024,    Adjusted R-squared:  -0.003176
## F-statistic: 0.7634 on 4 and 295 DF,  p-value: 0.5498
##
## Response variable: resid
## Total response variance: 0.03810211
## Analysis based on 300 observations
##
## 4 Regressors:
## my_trust_value my_reputation trust_value partner_reputation
## Proportion of variance explained by model: 1.02%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                           lmg        last      first       pratt
## my_trust_value     0.10006618 0.006757833 0.15553214 -0.04739675
## my_reputation      0.38876811 0.340069877 0.41951119  0.55163644
## trust_value        0.42016957 0.520160178 0.35261508  0.64287206
## partner_reputation 0.09099615 0.133012112 0.07234159 -0.14711175
##
## Average coefficients for different model sizes:
##
##                             1X         2Xs          3Xs          4Xs
## my_trust_value     0.03746181 0.019542732  0.004373675 -0.009742919
## my_reputation      0.06164771 0.065043649  0.067813320  0.069182831
## trust_value        0.05440987 0.063798595  0.075075558  0.084659041
## partner_reputation 0.02590801 0.002522776 -0.022569255 -0.044964048
## [1] "Analyze:  SENDER for game: ID GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation, data = cur_df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77299 -0.14399 -0.00552  0.16155  0.66015
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.18156    0.06279  -2.892  0.00412 **
## my_trust_value      0.40390    0.09201   4.390 1.58e-05 ***
## my_reputation      -0.33118    0.11521  -2.874  0.00434 **
## trust_value         0.45588    0.10334   4.412 1.44e-05 ***
## partner_reputation -0.07479    0.11992  -0.624  0.53332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2487 on 295 degrees of freedom
## Multiple R-squared:  0.1816, Adjusted R-squared:  0.1705
## F-statistic: 16.37 on 4 and 295 DF,  p-value: 4.021e-12
##
## Response variable: resid
## Total response variance: 0.07455148
## Analysis based on 300 observations
##
## 4 Regressors:
## my_trust_value my_reputation trust_value partner_reputation
## Proportion of variance explained by model: 18.16%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                          lmg       last        first        pratt
## my_trust_value     0.2857445 0.40666140 2.048281e-01  0.383150945
## my_reputation      0.0717870 0.17438407 6.739140e-06 -0.001442587
## trust_value        0.4923714 0.41074508 5.529255e-01  0.682233115
## partner_reputation 0.1500971 0.00820945 2.422396e-01 -0.063941473
##
## Average coefficients for different model sizes:
##
##                            1X        2Xs         3Xs        4Xs
## my_trust_value     0.27221514  0.3355681  0.36274149  0.4039006
## my_reputation      0.00195047 -0.1142068 -0.21586658 -0.3311758
## trust_value        0.46580470  0.4757269  0.46266381  0.4558815
## partner_reputation 0.35722099  0.2136565  0.06613895 -0.0747922
## [1] "Analyze:  SENDER for game: Score GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62122 -0.12345 -0.01111  0.14960  0.89020
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)        -0.23887    0.04820  -4.956 1.22e-06 ***
## my_trust_value      0.18190    0.07258   2.506   0.0127 *
## my_reputation      -0.49249    0.08685  -5.670 3.39e-08 ***
## trust_value         0.84316    0.07582  11.120  < 2e-16 ***
## partner_reputation  0.03975    0.09197   0.432   0.6659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2079 on 295 degrees of freedom
## Multiple R-squared:  0.4878, Adjusted R-squared:  0.4808
## F-statistic: 70.24 on 4 and 295 DF,  p-value: < 2.2e-16
##
## Response variable: resid
## Total response variance: 0.08329306
## Analysis based on 300 observations
##
## 4 Regressors:
## my_trust_value my_reputation trust_value partner_reputation
## Proportion of variance explained by model: 48.78%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                          lmg        last       first      pratt
## my_trust_value     0.05861385 0.038702887 0.067658609 0.06205836
## my_reputation      0.07007616 0.198140746 0.002118626 0.02399462
## trust_value        0.65232886 0.762005071 0.609181175 0.89028661
## partner_reputation 0.21898112 0.001151296 0.321041590 0.02366041
##
## Average coefficients for different model sizes:
##
##                             1X        2Xs        3Xs         4Xs
## my_trust_value      0.28509583  0.2057480  0.1618654  0.18189735
## my_reputation      -0.06251446 -0.3090957 -0.4122447 -0.49249120
## trust_value         0.82941012  0.8417100  0.8459051  0.84316195
## partner_reputation  0.77540469  0.5467906  0.2894768  0.03975097
## [1] "Analyze:  SENDER for game: Combine GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71289 -0.11355 -0.00793  0.08951  0.97398
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.12225    0.04616  -2.648  0.00852 **
## my_trust_value      0.43638    0.08371   5.213 3.50e-07 ***
## my_reputation      -0.59957    0.09183  -6.529 2.88e-10 ***
## trust_value         0.66962    0.08676   7.718 1.84e-13 ***
## partner_reputation -0.15211    0.09836  -1.546  0.12308
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.212 on 295 degrees of freedom
## Multiple R-squared:  0.3744, Adjusted R-squared:  0.3659
## F-statistic: 44.14 on 4 and 295 DF,  p-value: < 2.2e-16
##
## Response variable: resid
## Total response variance: 0.07089466
## Analysis based on 300 observations
##
## 4 Regressors:
## my_trust_value my_reputation trust_value partner_reputation
## Proportion of variance explained by model: 37.44%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                          lmg       last       first       pratt
## my_trust_value     0.1808145 0.20625023 0.165421246  0.25683563
## my_reputation      0.1411515 0.32354218 0.002892745  0.04012743
## trust_value        0.5346637 0.45205989 0.590782152  0.79896151
## partner_reputation 0.1433703 0.01814771 0.240903858 -0.09592458
##
## Average coefficients for different model sizes:
##
##                             1X        2Xs         3Xs        4Xs
## my_trust_value      0.34747155  0.3550695  0.38505055  0.4363785
## my_reputation      -0.05343548 -0.3425072 -0.50168998 -0.5995714
## trust_value         0.61213909  0.6567763  0.68419013  0.6696204
## partner_reputation  0.47225613  0.2396086  0.01271427 -0.1521051
## [1] "Analyze:  RECEIVER for game: Simple GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation + AbsPartnerSend, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58659 -0.05810 -0.00234  0.06802  0.56640
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.016760   0.036735  -0.456    0.649
## my_trust_value      0.095196   0.081074   1.174    0.242
## my_reputation      -0.081490   0.089691  -0.909    0.365
## trust_value         0.017274   0.085332   0.202    0.840
## partner_reputation  0.044859   0.095766   0.468    0.640
## AbsPartnerSend     -0.002220   0.006074  -0.366    0.715
##
## Residual standard error: 0.1723 on 184 degrees of freedom
## Multiple R-squared:  0.01026,    Adjusted R-squared:  -0.01664
## F-statistic: 0.3814 on 5 and 184 DF,  p-value: 0.8611
##
## Response variable: resid
```

```
## Total response variance: 0.02921454
## Analysis based on 190 observations
##
## 5 Regressors:
## my_trust_value my_reputation trust_value partner_reputation AbsPartnerSend
## Proportion of variance explained by model: 1.03%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                           lmg       last       first       pratt
## my_trust_value     0.47809991 0.53064153 0.372242771  0.61837743
## my_reputation      0.21978620 0.31771273 0.009530741  0.07634450
## trust_value        0.09050034 0.01577107 0.246243338  0.08051691
## partner_reputation 0.17596724 0.08445129 0.360578282  0.25172767
## AbsPartnerSend     0.03564631 0.05142338 0.011404868 -0.02696652
##
## Average coefficients for different model sizes:
##
##                              1X           2Xs          3Xs         4Xs
## my_trust_value      0.034853478  0.0502119128  0.065526556  0.08069888
## my_reputation      -0.006187369 -0.0239379962 -0.042545180 -0.06197499
## trust_value         0.032130108  0.0280722495  0.023895581  0.02030411
## partner_reputation  0.039081697  0.0415568341  0.043950655  0.04535616
## AbsPartnerSend      0.000571145 -0.0003448097 -0.001150051 -0.00181561
##                             5Xs
## my_trust_value      0.095196087
## my_reputation      -0.081489699
## trust_value         0.017273503
## partner_reputation  0.044859087
## AbsPartnerSend     -0.002220377
## [1] "Analyze:  RECEIVER for game: ID GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation + AbsPartnerSend, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52846 -0.06044 -0.00865  0.07603  0.55658
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.127243   0.047042  -2.705  0.00728 **
## my_trust_value      0.031023   0.078183   0.397  0.69184
## my_reputation       0.006915   0.086981   0.080  0.93669
## trust_value        -0.034696   0.067922  -0.511  0.60991
## partner_reputation  0.265332   0.089318   2.971  0.00325 **
## AbsPartnerSend     -0.000928   0.004208  -0.221  0.82564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1638 on 261 degrees of freedom
## Multiple R-squared:  0.05026,    Adjusted R-squared:  0.03207
```

```
## F-statistic: 2.763 on 5 and 261 DF,  p-value: 0.01884
##
## Response variable: resid
## Total response variance: 0.02771811
## Analysis based on 267 observations
##
## 5 Regressors:
## my_trust_value my_reputation trust_value partner_reputation AbsPartnerSend
## Proportion of variance explained by model: 5.03%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                            lmg         last       first        pratt
## my_trust_value     0.011148705 0.0169336657 0.005692923  0.015232178
## my_reputation      0.006919924 0.0006798383 0.001245479 -0.001371862
## trust_value        0.137715310 0.0280640021 0.212326960 -0.106744808
## partner_reputation 0.771571195 0.9490928871 0.650391549  1.125836036
## AbsPartnerSend     0.072644866 0.0052296069 0.130343089 -0.032951545
##
## Average coefficients for different model sizes:
##
##                              1X         2Xs          3Xs           4Xs
## my_trust_value      0.017057575  0.01547853  0.017106053  2.358795e-02
## my_reputation      -0.009236620 -0.01712939 -0.004250897  4.645363e-03
## trust_value         0.101531853  0.06421763  0.026945807 -5.931685e-03
## partner_reputation  0.225502117  0.23338945  0.244872282  2.561044e-01
## AbsPartnerSend      0.005400058  0.00348313  0.001604092  7.380727e-05
##                              5Xs
## my_trust_value      0.0310229079
## my_reputation       0.0069155301
## trust_value        -0.0346962434
## partner_reputation  0.2653318636
## AbsPartnerSend     -0.0009279503
## [1] "Analyze:  RECEIVER for game: Score GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation + AbsPartnerSend, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52421 -0.06767  0.00818  0.07518  0.50831
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.066898   0.047574  -1.406 0.160928
## my_trust_value      0.039421   0.078932   0.499 0.617924
## my_reputation       0.013624   0.081557   0.167 0.867468
## trust_value         0.256980   0.065083   3.949 0.000103 ***
## partner_reputation  0.043975   0.082612   0.532 0.594988
## AbsPartnerSend     -0.015698   0.005077  -3.092 0.002217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1625 on 246 degrees of freedom
## Multiple R-squared:  0.09111,    Adjusted R-squared:  0.07264
## F-statistic: 4.932 on 5 and 246 DF,  p-value: 0.0002543
##
## Response variable: resid
## Total response variance: 0.02845779
## Analysis based on 252 observations
##
## 5 Regressors:
## my_trust_value my_reputation trust_value partner_reputation AbsPartnerSend
## Proportion of variance explained by model: 9.11%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                          lmg        last       first        pratt
## my_trust_value     0.02966988 0.009700866 0.005984491 -0.009493285
## my_reputation      0.01147320 0.001085321 0.023799432 -0.005117618
## trust_value        0.60613892 0.606365072 0.750416848  0.705796731
## partner_reputation 0.07290413 0.011020538 0.104805769  0.037386188
## AbsPartnerSend     0.27981387 0.371828202 0.114993460  0.271427984
##
## Average coefficients for different model sizes:
##
##                             1X          2Xs           3Xs          4Xs
## my_trust_value     -0.014697544 -0.017500965 -0.0008732984  0.021121877
## my_reputation      -0.037472496 -0.040944482 -0.0157694546  0.004574975
## trust_value         0.161595176  0.210990260  0.2426588364  0.256242582
## partner_reputation  0.072910470  0.064327493  0.0558279598  0.049848924
## AbsPartnerSend     -0.003933333 -0.008013429 -0.0118663949 -0.014457661
##                            5Xs
## my_trust_value      0.03942107
## my_reputation       0.01362411
## trust_value         0.25698024
## partner_reputation  0.04397544
## AbsPartnerSend     -0.01569772
## [1] "Analyze:  RECEIVER for game: Combine GAME"
##
## Call:
## lm(formula = resid ~ my_trust_value + my_reputation + trust_value +
##     partner_reputation + AbsPartnerSend, data = cur_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47853 -0.05522  0.00090  0.06589  0.78251
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.071235   0.036269  -1.964   0.0506 .
## my_trust_value     -0.018032   0.068805  -0.262   0.7935
## my_reputation      -0.031016   0.070882  -0.438   0.6621
## trust_value         0.288056   0.065974   4.366 1.85e-05 ***
## partner_reputation -0.043111   0.071325  -0.604   0.5461
```

```
## AbsPartnerSend     -0.002536   0.003905  -0.649   0.5167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.141 on 252 degrees of freedom
## Multiple R-squared:  0.1051, Adjusted R-squared:  0.08734
## F-statistic: 5.919 on 5 and 252 DF,  p-value: 3.409e-05
##
## Response variable: resid
## Total response variance: 0.02178139
## Analysis based on 258 observations
##
## 5 Regressors:
## my_trust_value my_reputation trust_value partner_reputation AbsPartnerSend
## Proportion of variance explained by model: 10.51%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                          lmg        last       first        pratt
## my_trust_value     0.01162571 0.003415011 0.005217417 -0.006495888
## my_reputation      0.01890928 0.009520442 0.003536607  0.007985521
## trust_value        0.76400617 0.947928200 0.661385748  1.157620200
## partner_reputation 0.14435930 0.018165543 0.234063758 -0.094071690
## AbsPartnerSend     0.06109955 0.020970803 0.095796470 -0.065038142
##
## Average coefficients for different model sizes:
##
##                             1X         2Xs          3Xs          4Xs
## my_trust_value      0.019782043 -0.01025073 -0.017810175 -0.016212989
## my_reputation      -0.018762286 -0.05961094 -0.053685148 -0.043231864
## trust_value         0.224795939  0.25173556  0.268500166  0.280042370
## partner_reputation  0.146515430  0.09506225  0.043689528 -0.002303299
## AbsPartnerSend      0.005101693  0.00232843 -0.000109255 -0.001776741
##                            5Xs
## my_trust_value     -0.018031508
## my_reputation      -0.031015595
## trust_value         0.288056475
## partner_reputation -0.043110797
## AbsPartnerSend     -0.002535776
```

Analyze effect of SHOW_TRUST and SHOW_ID in individual send proportion

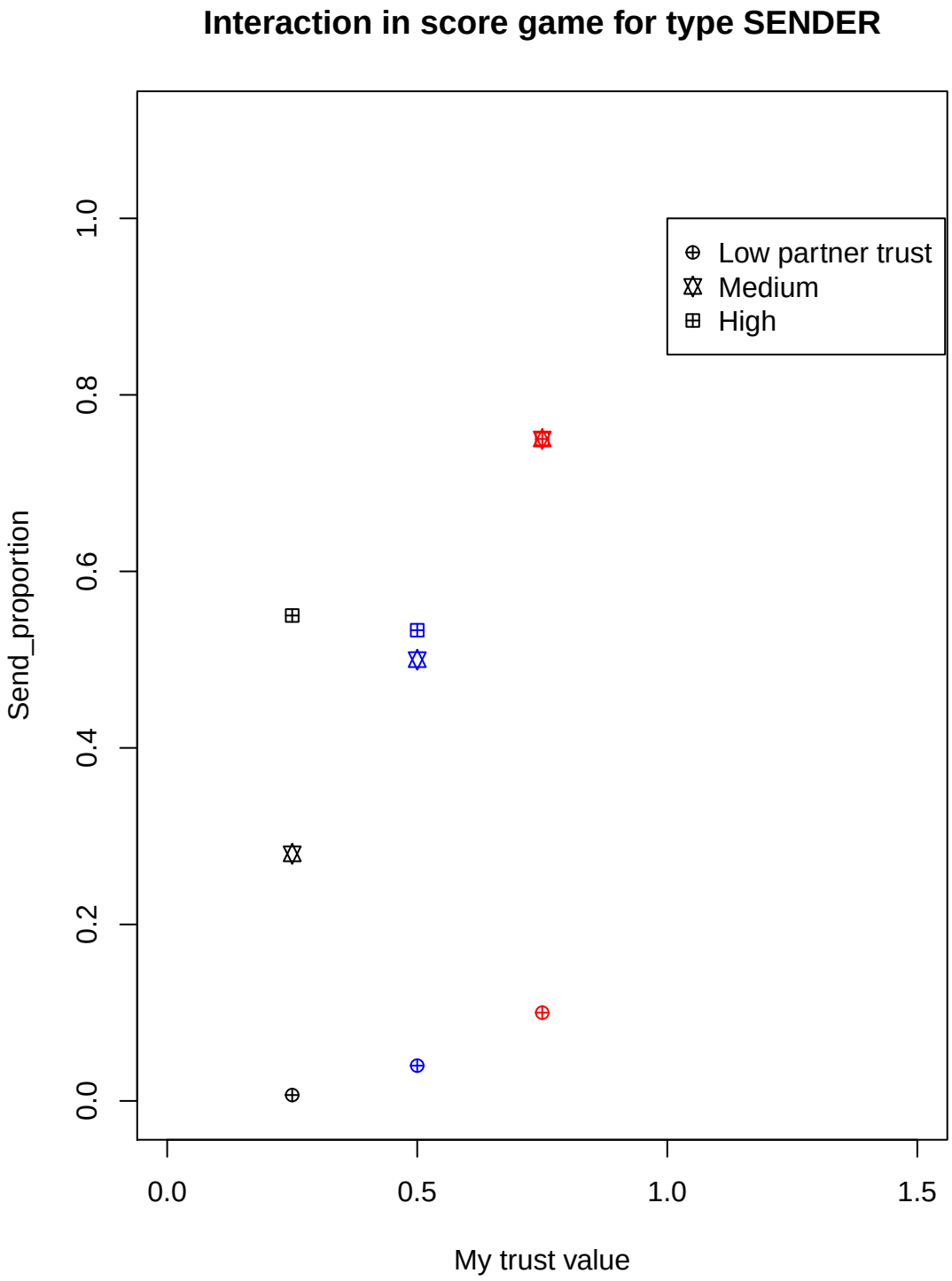
Date: Wed 14-Jun-2017

We will perform ANOVA without groupID.

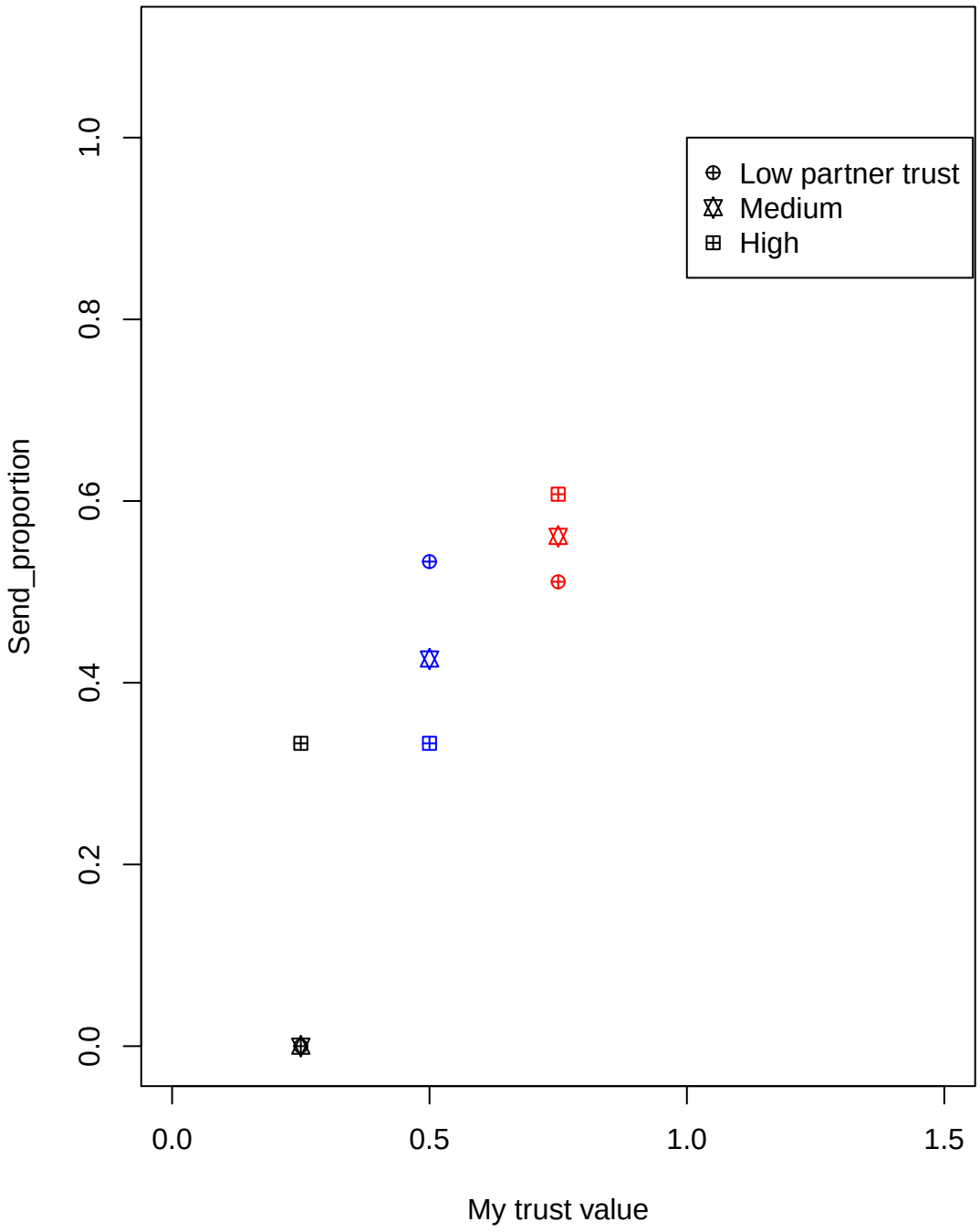
```
## [1] "ANOVA on SHOW_TRUST and SHOW_ID for: SENDER"
##
## Error: id
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1  4.318   4.318
##
## Error: id:SHOW_TRUST
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1   2.97    2.97
##
## Error: id:SHOW_ID
##         Df Sum Sq Mean Sq
## SHOW_ID  1  3.307   3.307
##
## Error: id:SHOW_TRUST:SHOW_ID
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1   1.13    1.13
##
## Error: Within
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## SHOW_TRUST             1   3.08   3.076   26.61 2.83e-07 ***
## SHOW_ID                1   2.72   2.725   23.57 1.33e-06 ***
## SHOW_TRUST:SHOW_ID     1   5.39   5.394   46.65 1.23e-11 ***
## Residuals           1492 172.50   0.116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "ANOVA on SHOW_TRUST and SHOW_ID for: RECEIVER"
##
## Error: id
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1 0.3324  0.3324
##
## Error: id:SHOW_TRUST
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1   2.85    2.85
##
## Error: id:SHOW_ID
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1  1.094   1.094
##
## Error: id:SHOW_TRUST:SHOW_ID
##            Df Sum Sq Mean Sq
## SHOW_TRUST  1  1.345   1.345
##
## Error: Within
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## SHOW_TRUST             1   1.01   1.014   19.51 1.09e-05 ***
## SHOW_ID                1   1.38   1.380   26.57 2.96e-07 ***
## SHOW_TRUST:SHOW_ID     1   1.26   1.256   24.17 1.00e-06 ***
```

```
## Residuals          1235  64.16   0.052
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

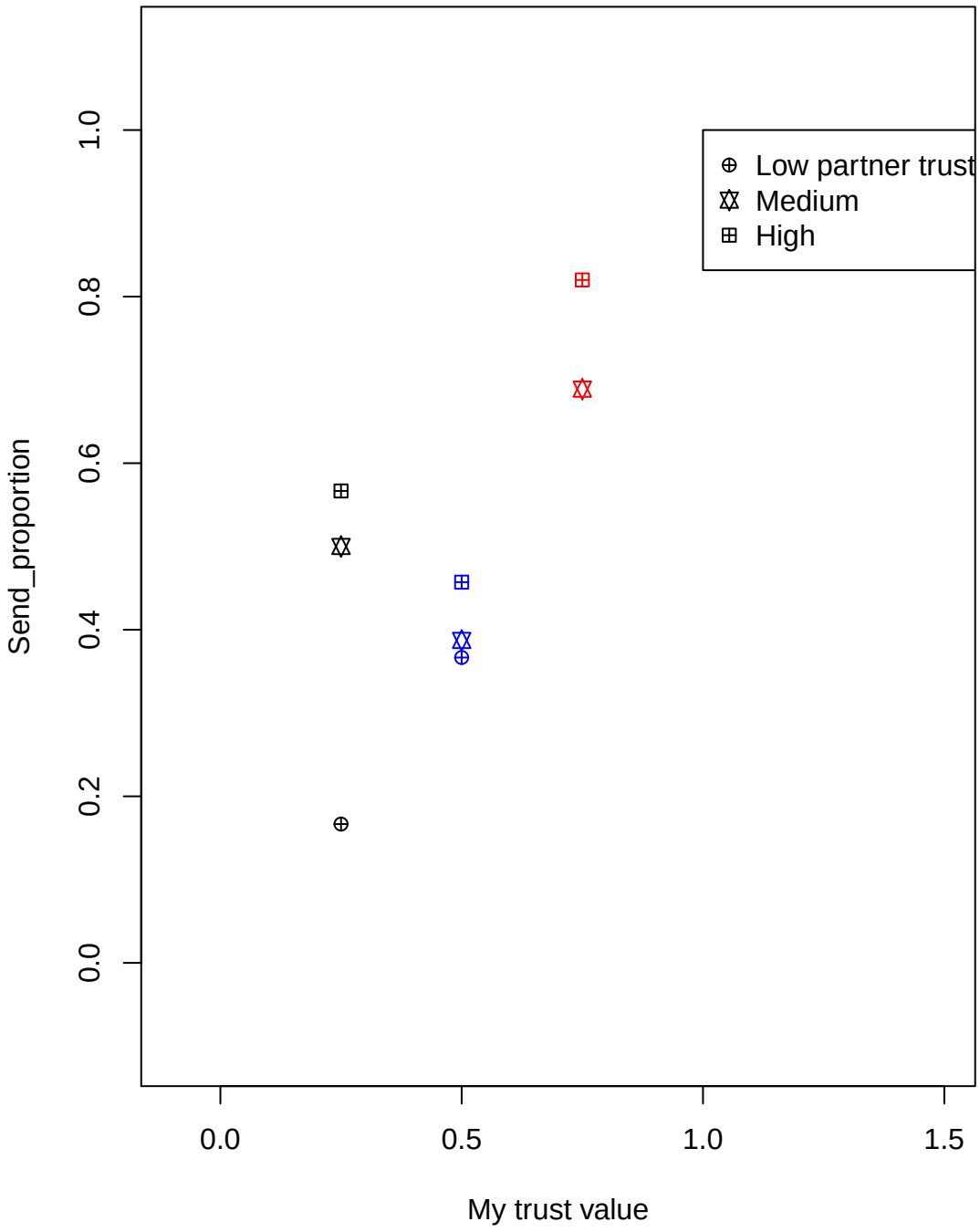
Understanding interactions between two trust values



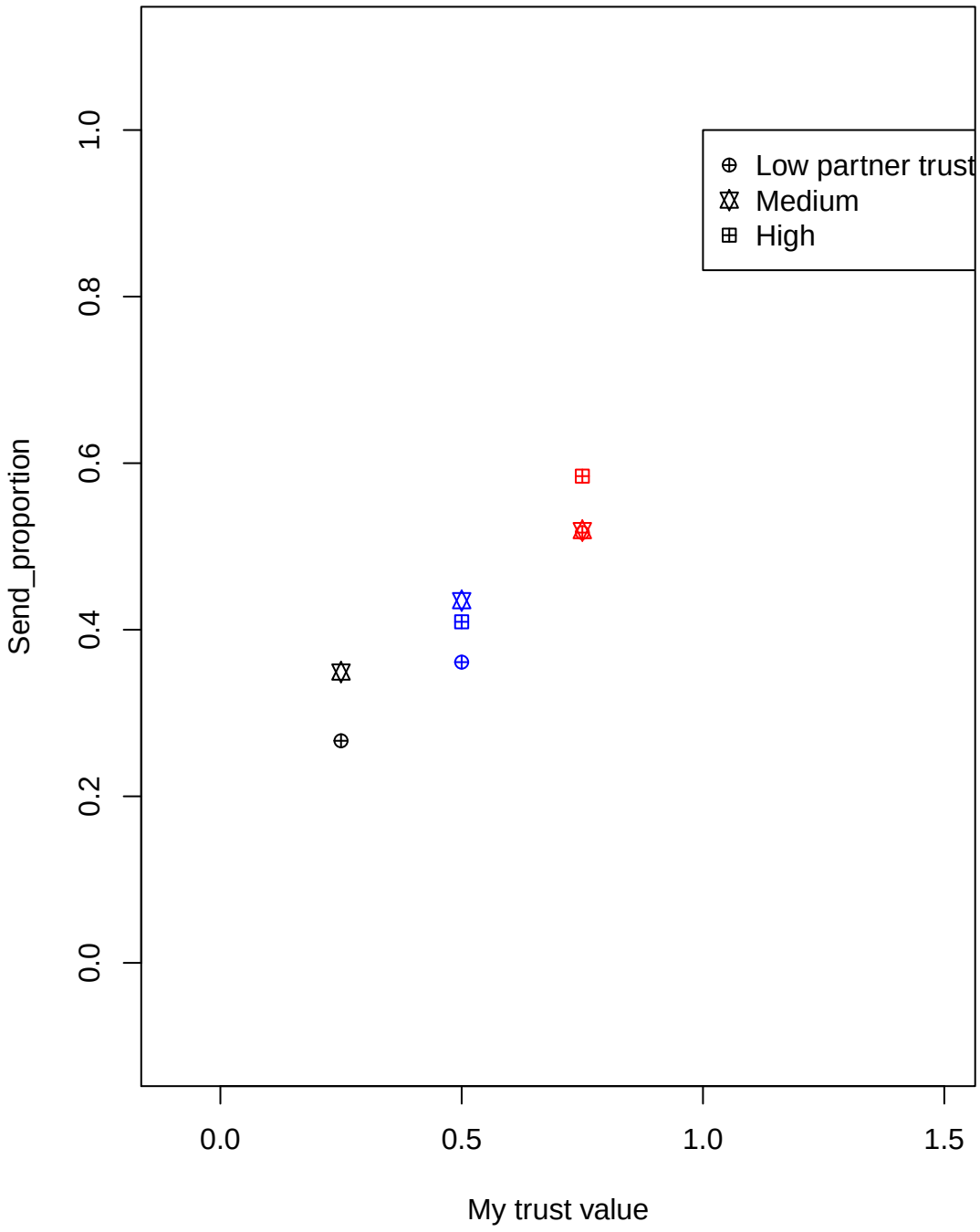

```
##
##
## \pagebreak
```


Interaction in score game for type RECEIVER
Send_proportion
My trust value
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
Low partner trust
Medium
High

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


Interaction in combine game for type SENDER
Send_proportion
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
My trust value
Low partner trust
Medium
High

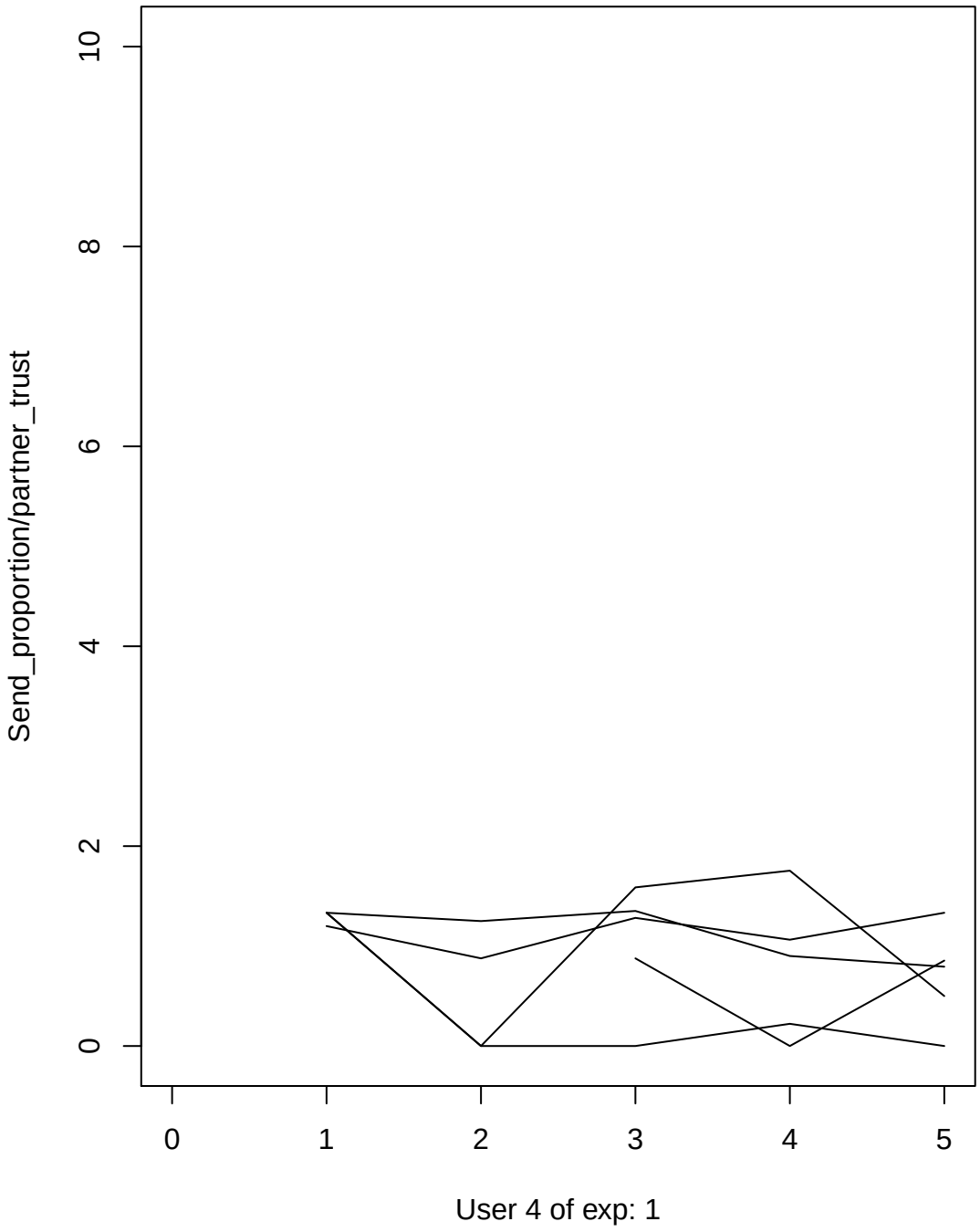
```
##
##
## \pagebreak
```


Interaction in combine game for type RECEIVER
Send_proportion
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
My trust value
Low partner trust
Medium
High

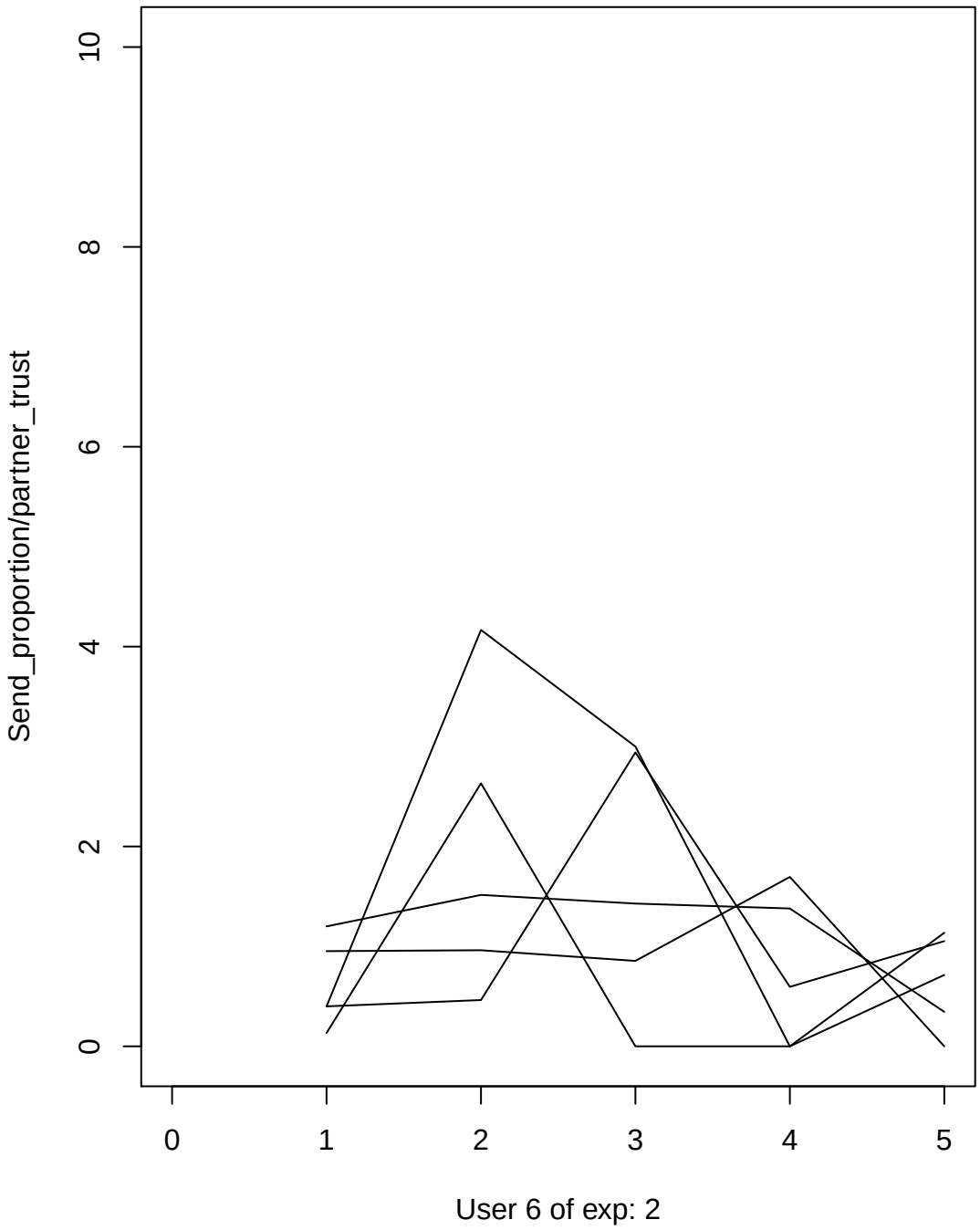
```
##
##
## \pagebreak
```

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

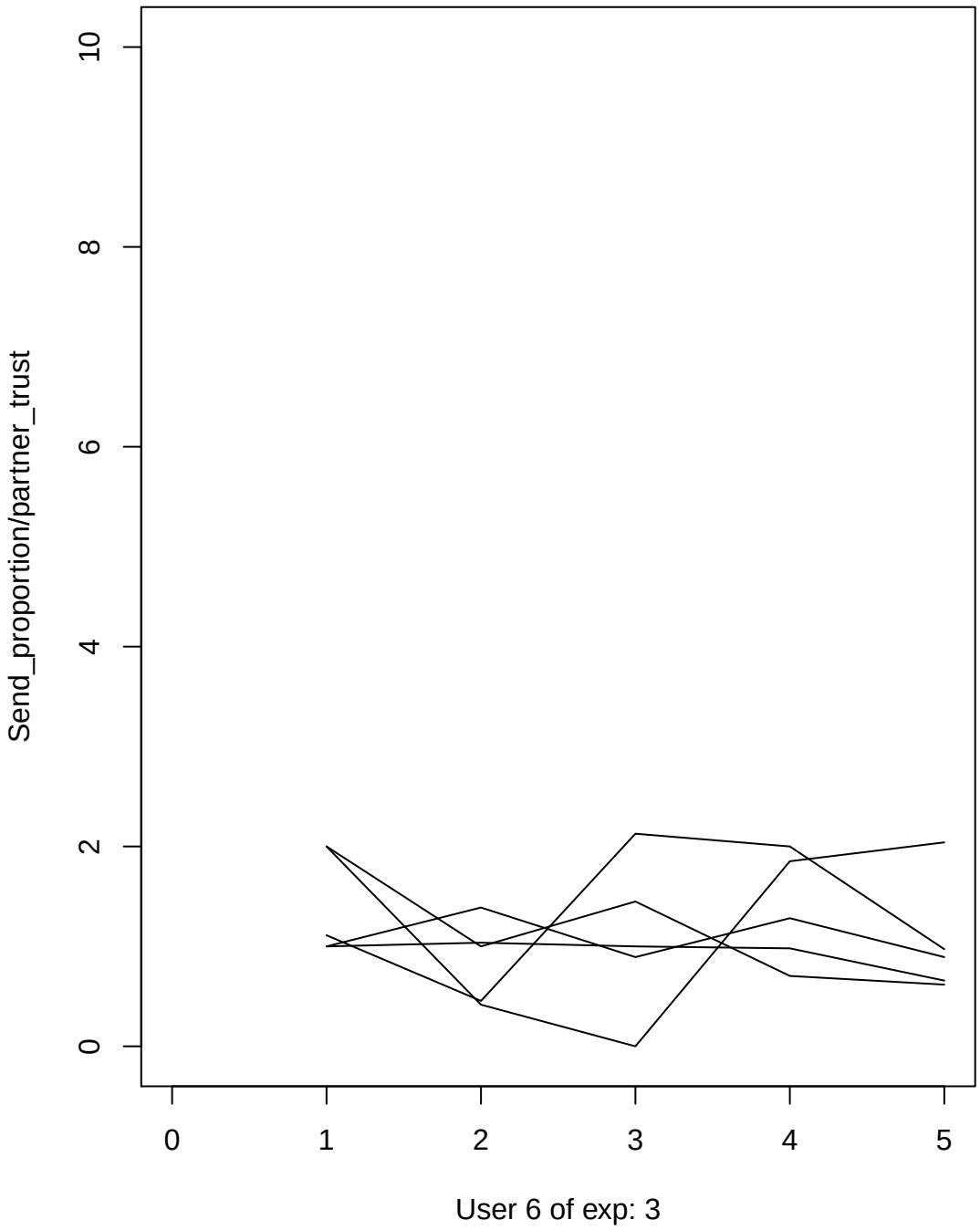
In this section, we will see do the behavior of one user converge in the end of a game.



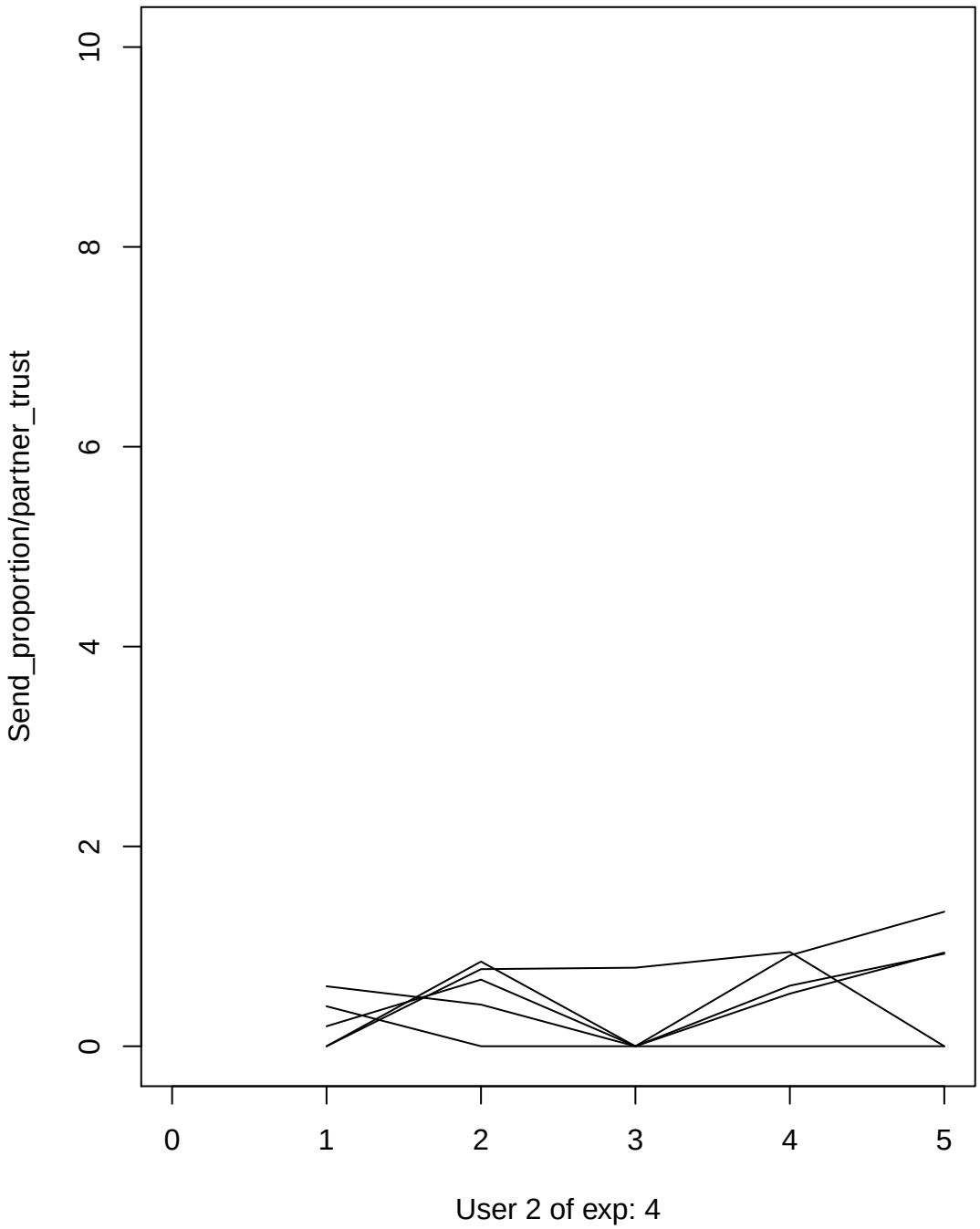

```
##
##
## \pagebreak
```


Send_proportion/partner_trust
0
2
4
6
8
10
0
1
2
3
4
5
User 6 of exp: 2

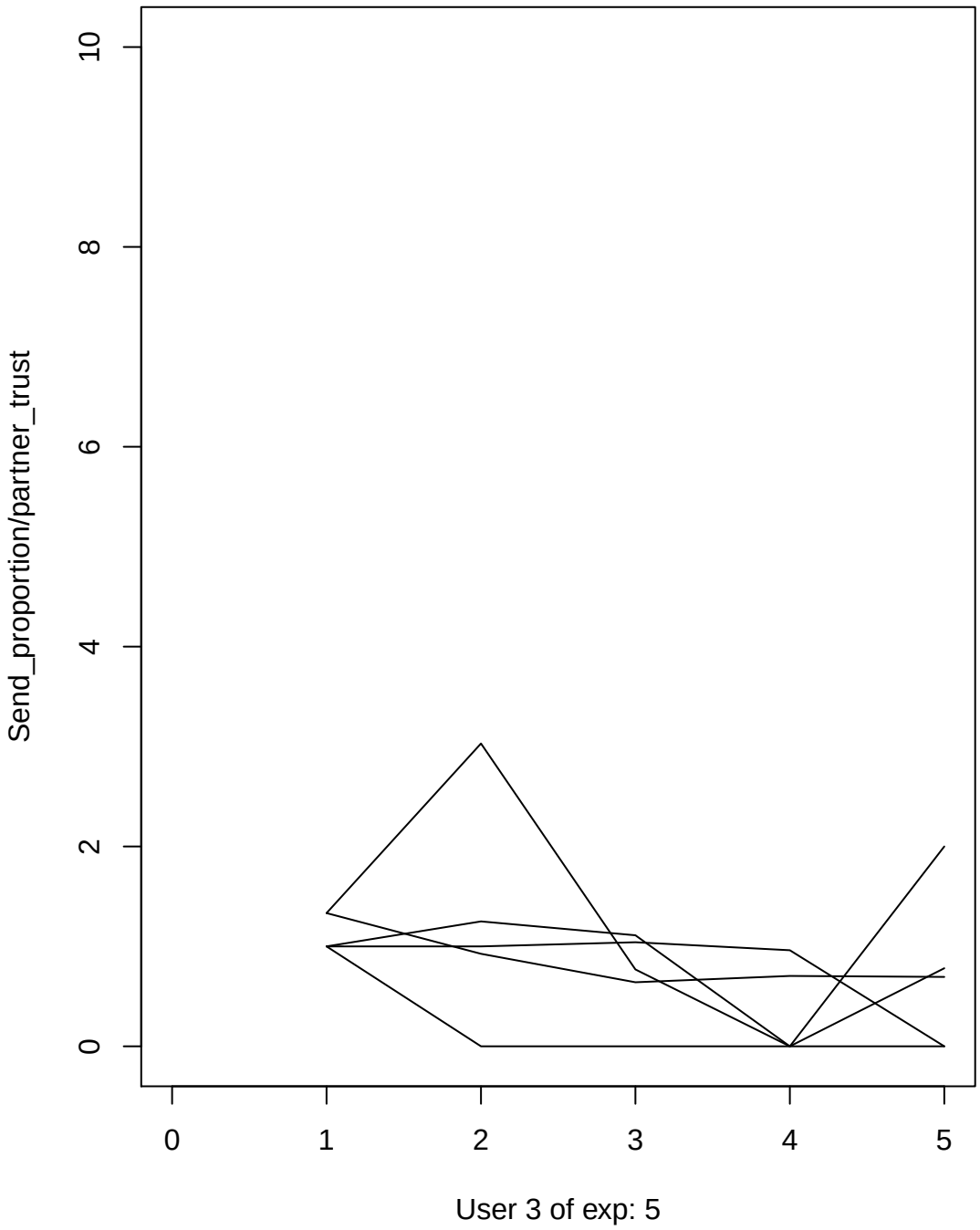
```
##
##
## \pagebreak
```


Send_proportion/partner_trust
0
2
4
6
8
10
0
1
2
3
4
5
User 6 of exp: 3

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

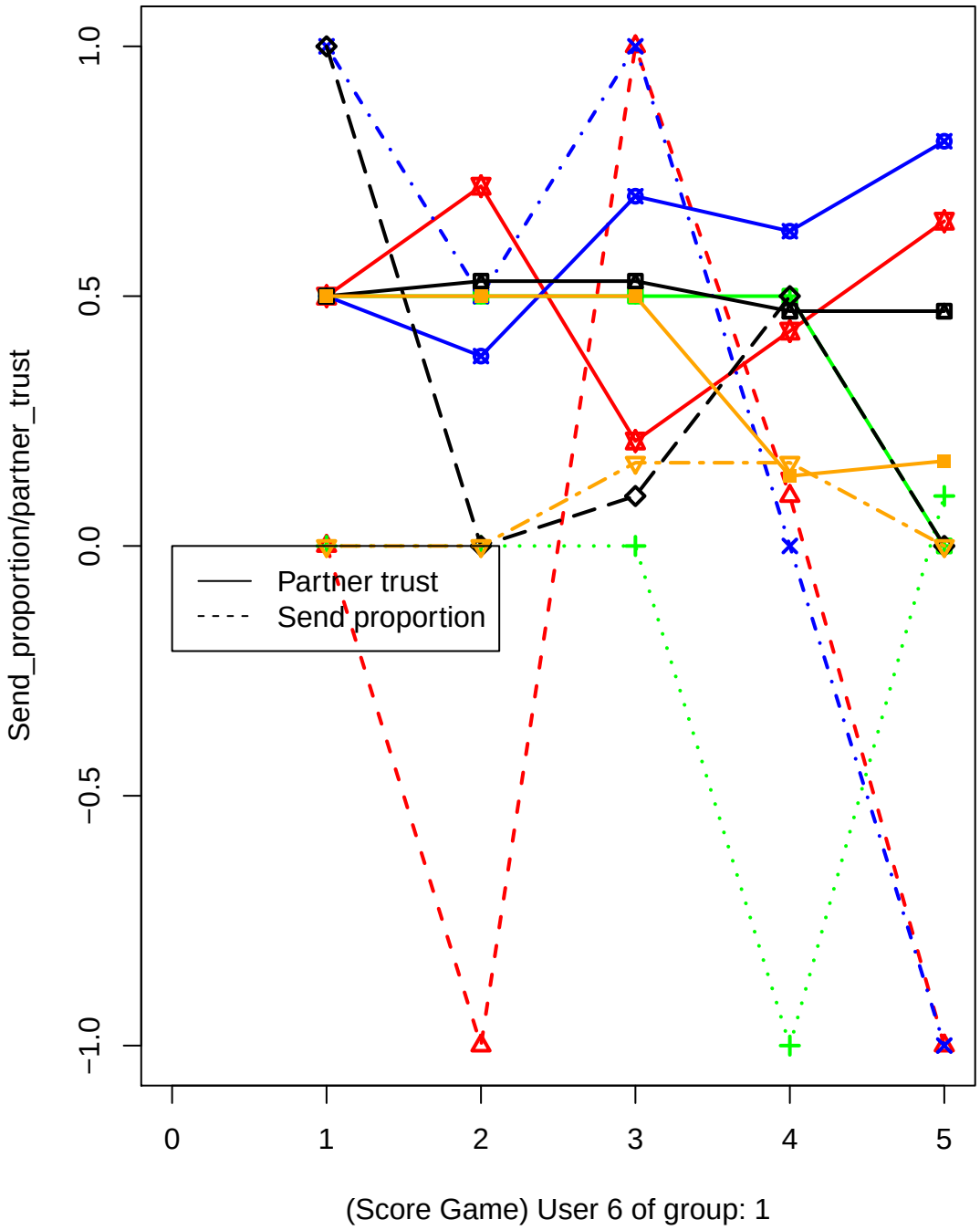

Send_proportion/partner_trust
0
2
4
6
8
10
0
1
2
3
4
5
User 2 of exp: 4

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

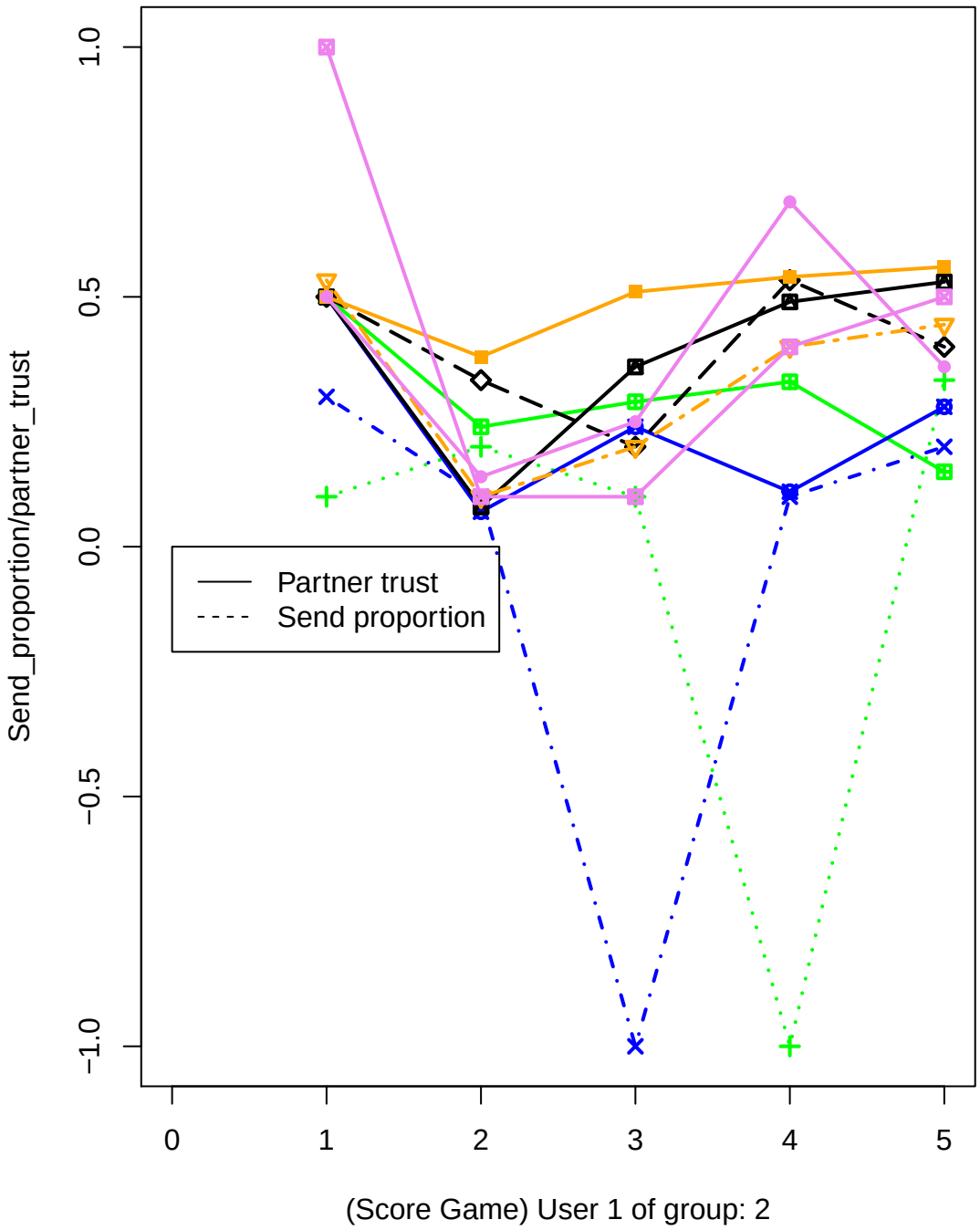

Send_proportion/partner_trust
0
2
4
6
8
10
0
1
2
3
4
5
User 3 of exp: 5

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

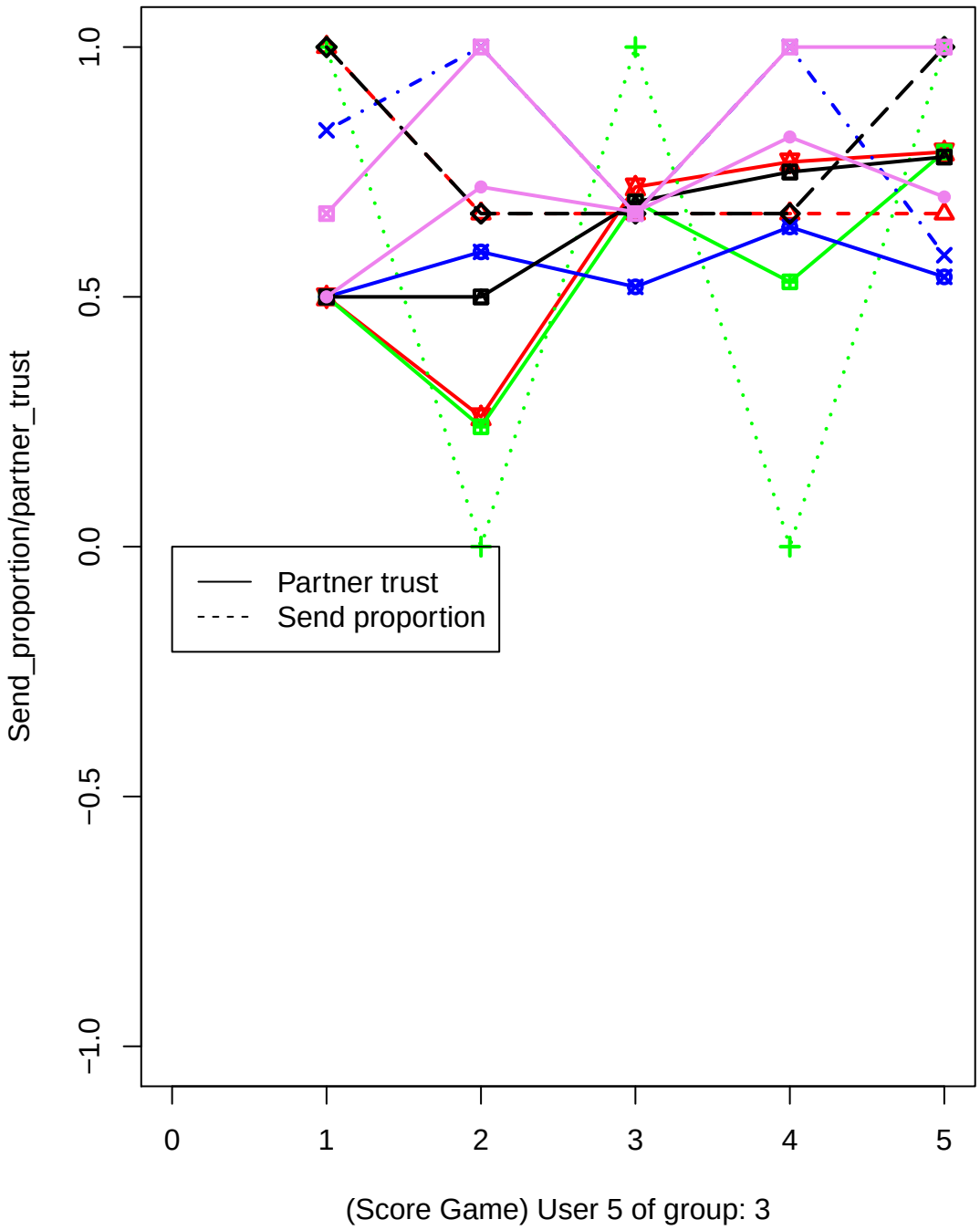
As the metric send_proportion/partner_trust does not say anything, we now display both send_proportion and partner_trust in one graph to check the stability of user behavior over time.



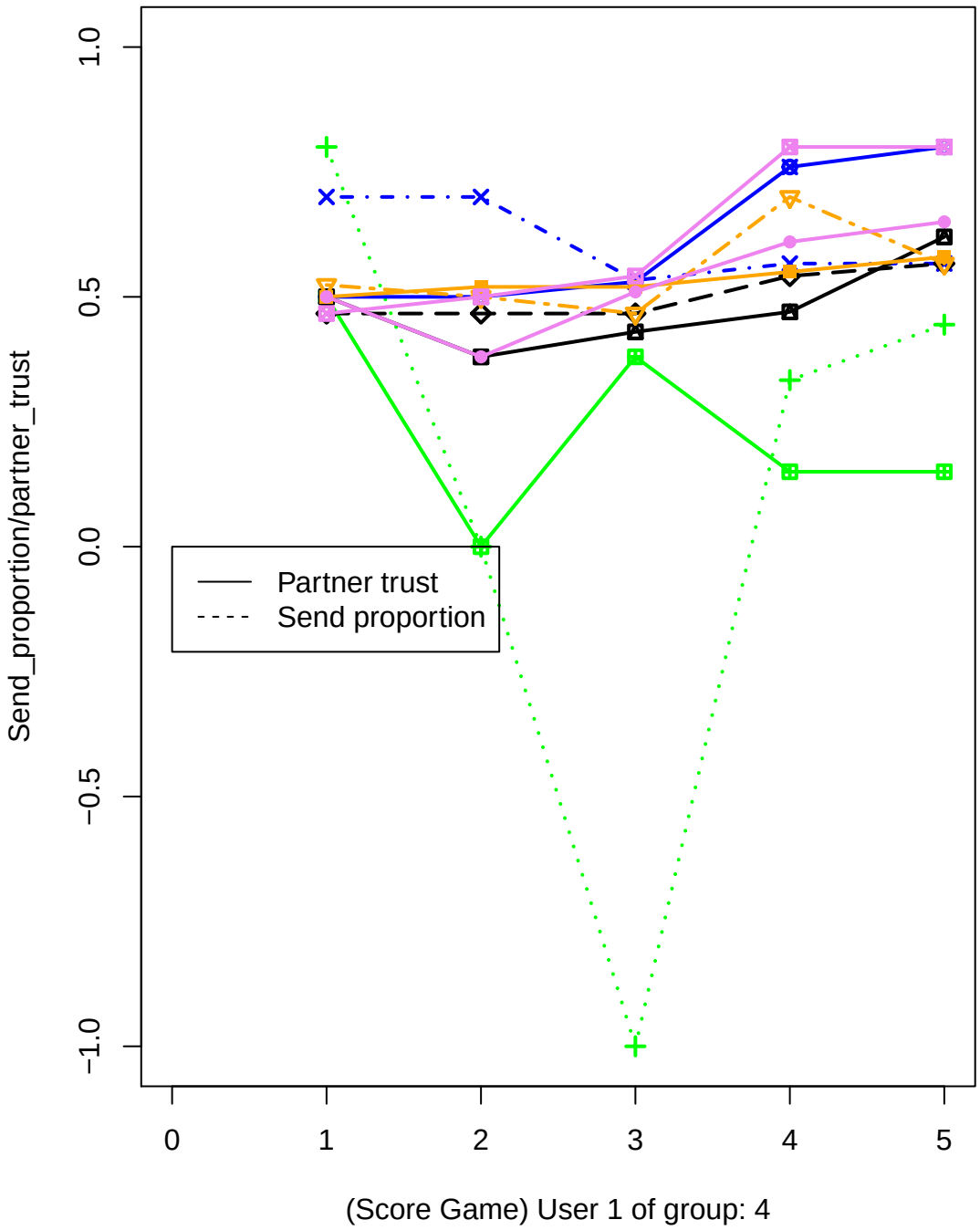

```
##
##
## \pagebreak
```


Send_proportion/partner_trust
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
Partner trust
Send proportion
(Score Game) User 1 of group: 2

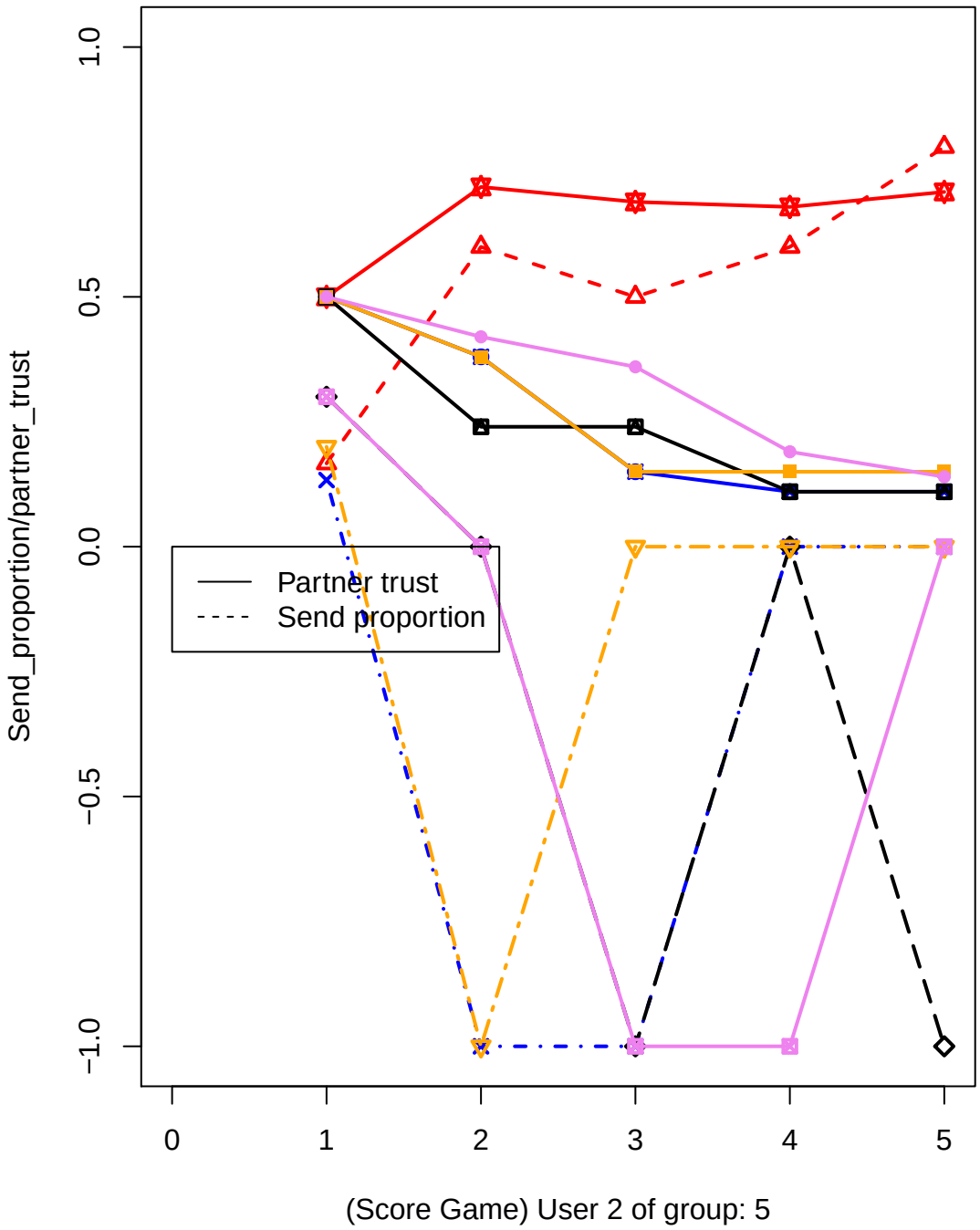
```
##
##
## \pagebreak
```


Send_proportion/partner_trust
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
Partner trust
Send proportion
(Score Game) User 5 of group: 3

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


Send_proportion/partner_trust
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
Partner trust
Send proportion
(Score Game) User 1 of group: 4

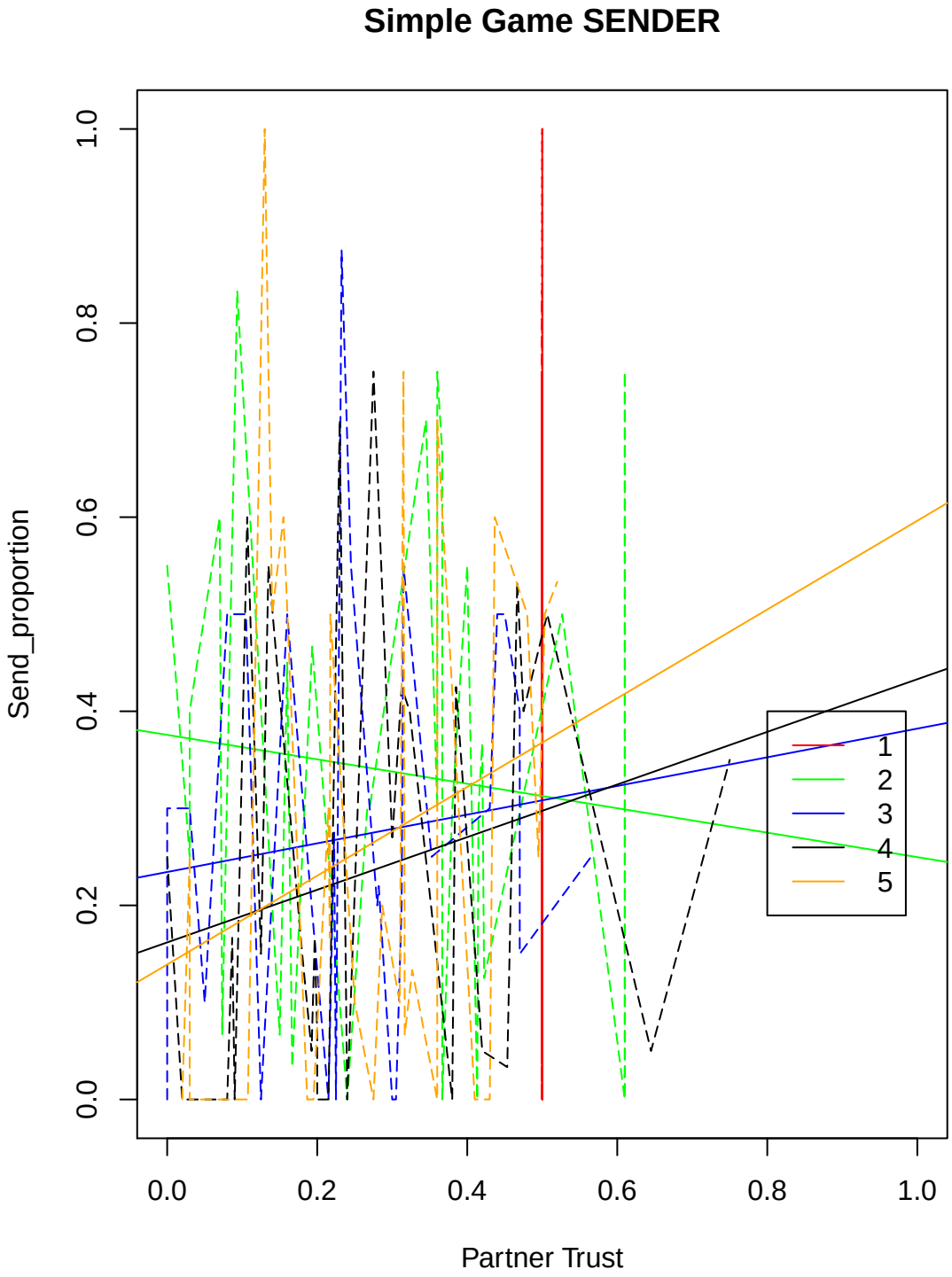
```
##
##
## \pagebreak
```


Send_proportion/partner_trust
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
Partner trust
Send proportion
(Score Game) User 2 of group: 5

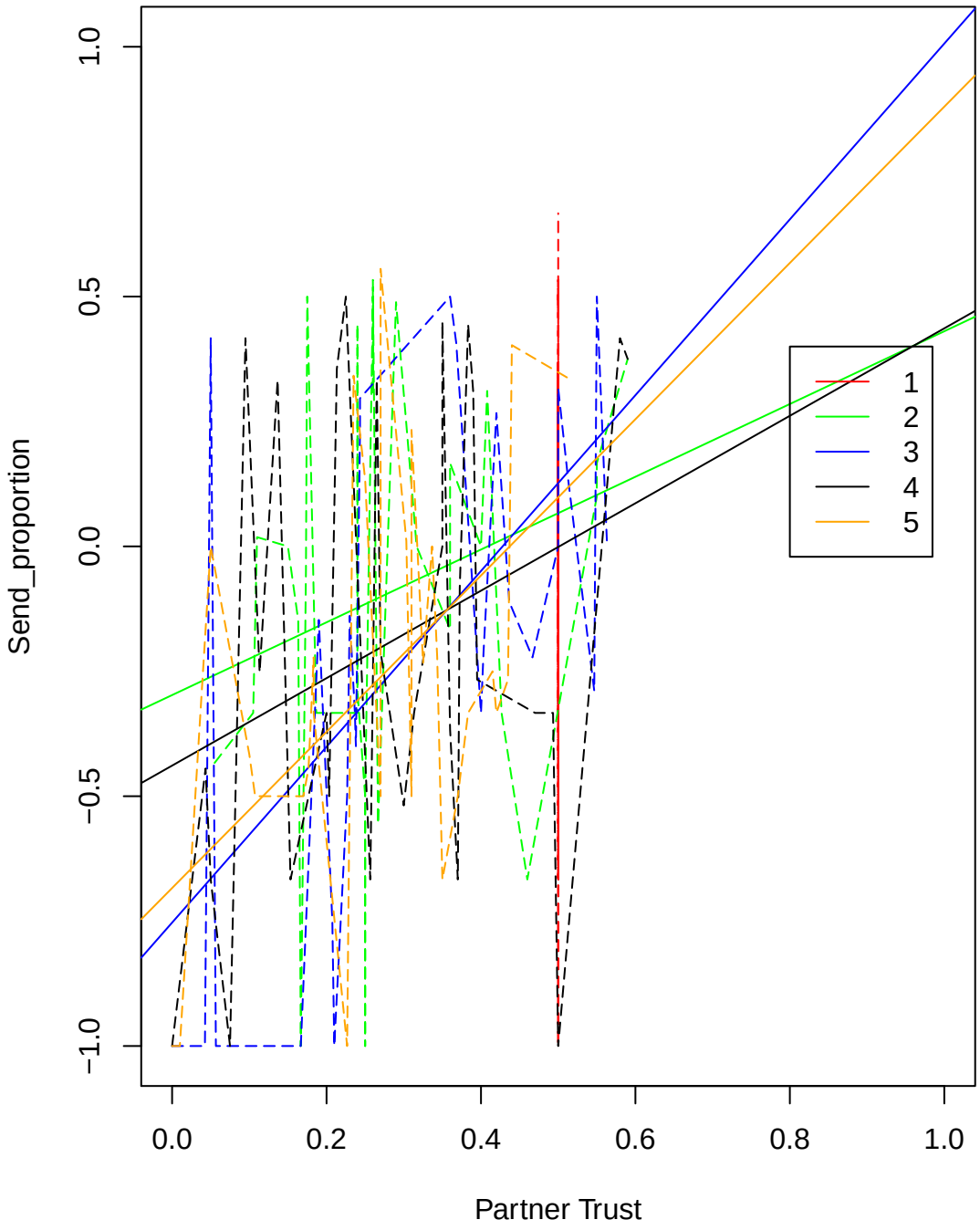
```
##
##
## \pagebreak
```

Change of send_proportion ~ partner_trust over time

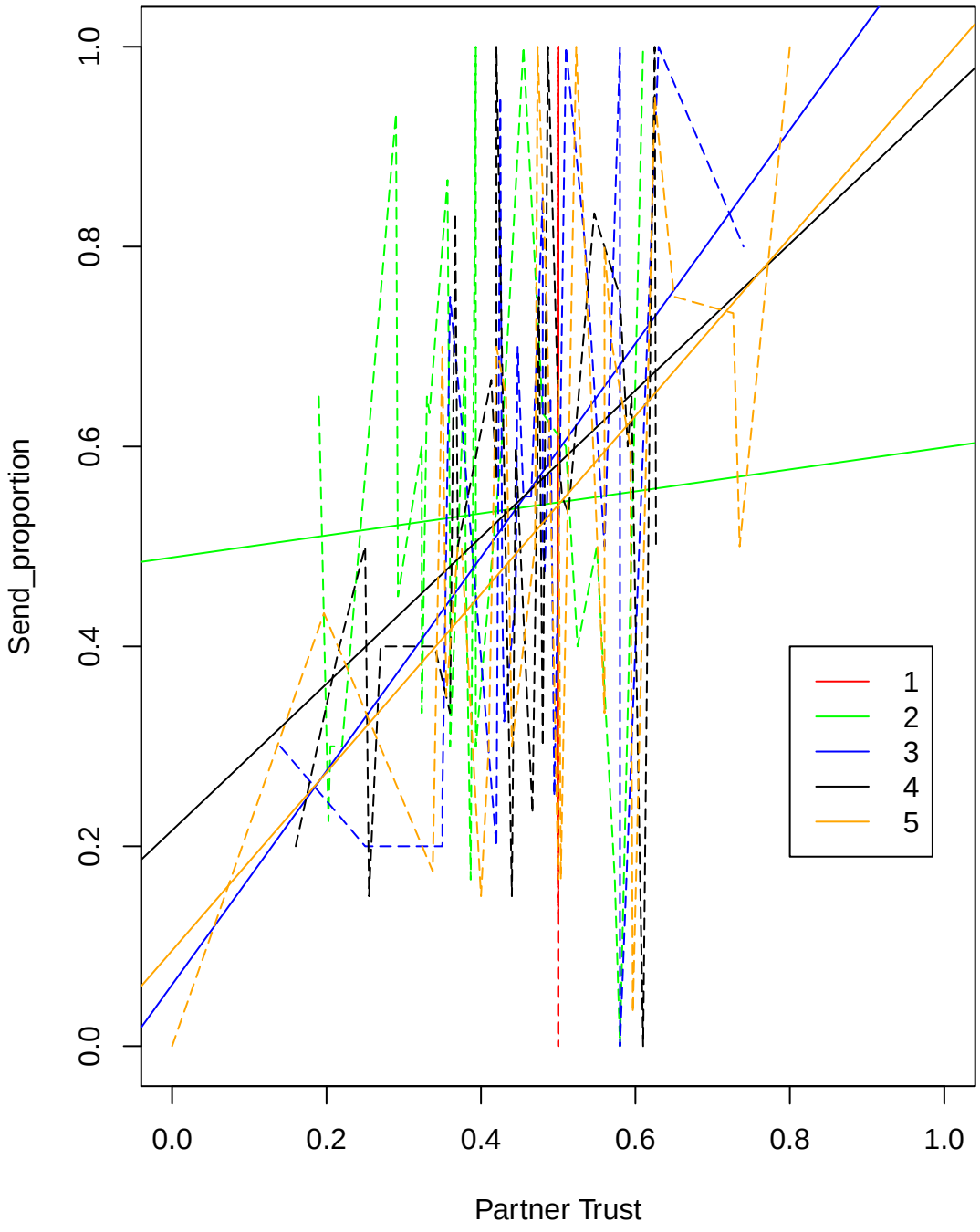
We want to see how user send proportion depends on partner trust in each round for each game



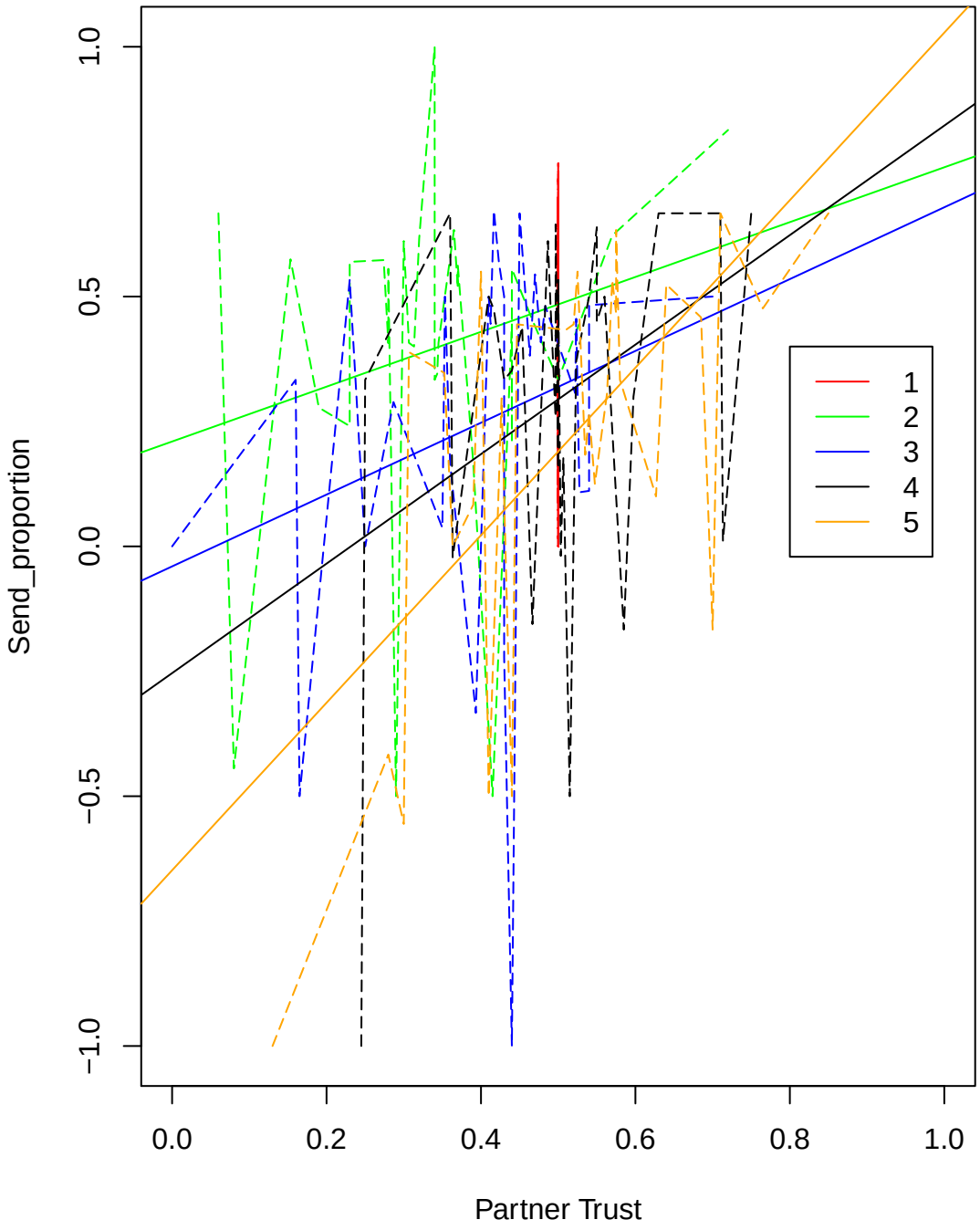

```
##
##
## \pagebreak
```


Simple Game RECEIVER
Send_proportion
Partner Trust
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

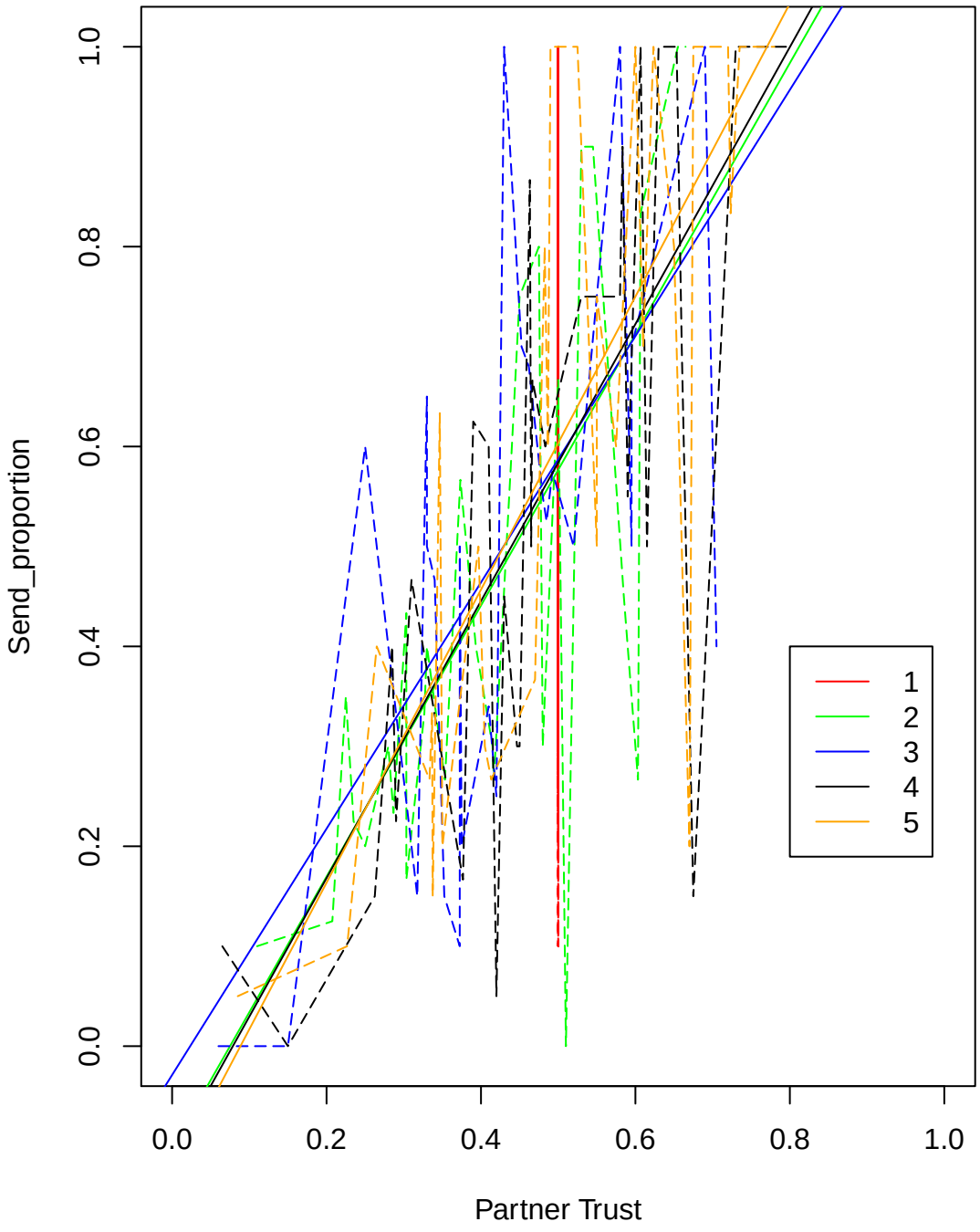
```
##
##
## \pagebreak
```


ID Game SENDER
Send_proportion
Partner Trust
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

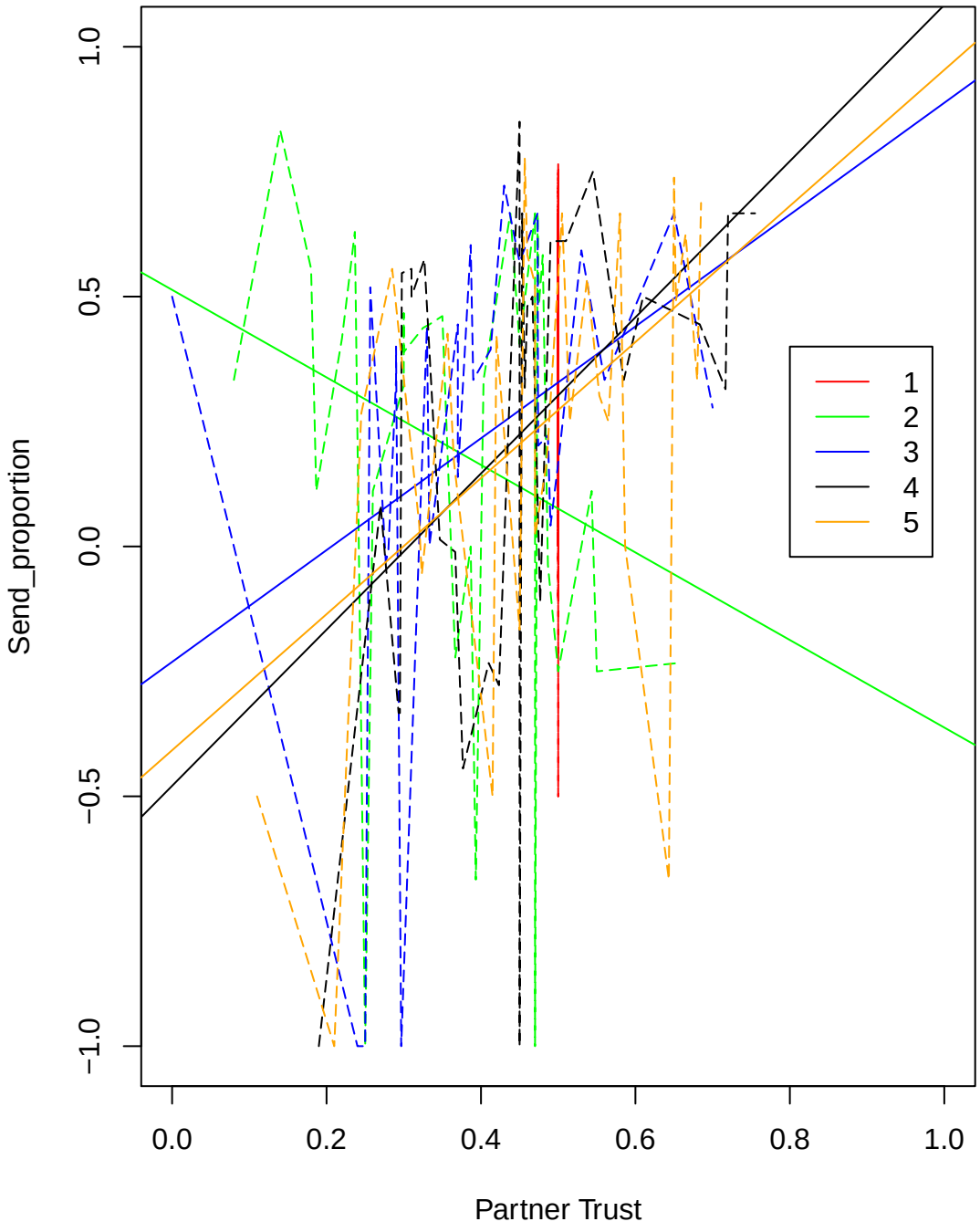
```
##
##
## \pagebreak
```


ID Game RECEIVER
Send_proportion
Partner Trust
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

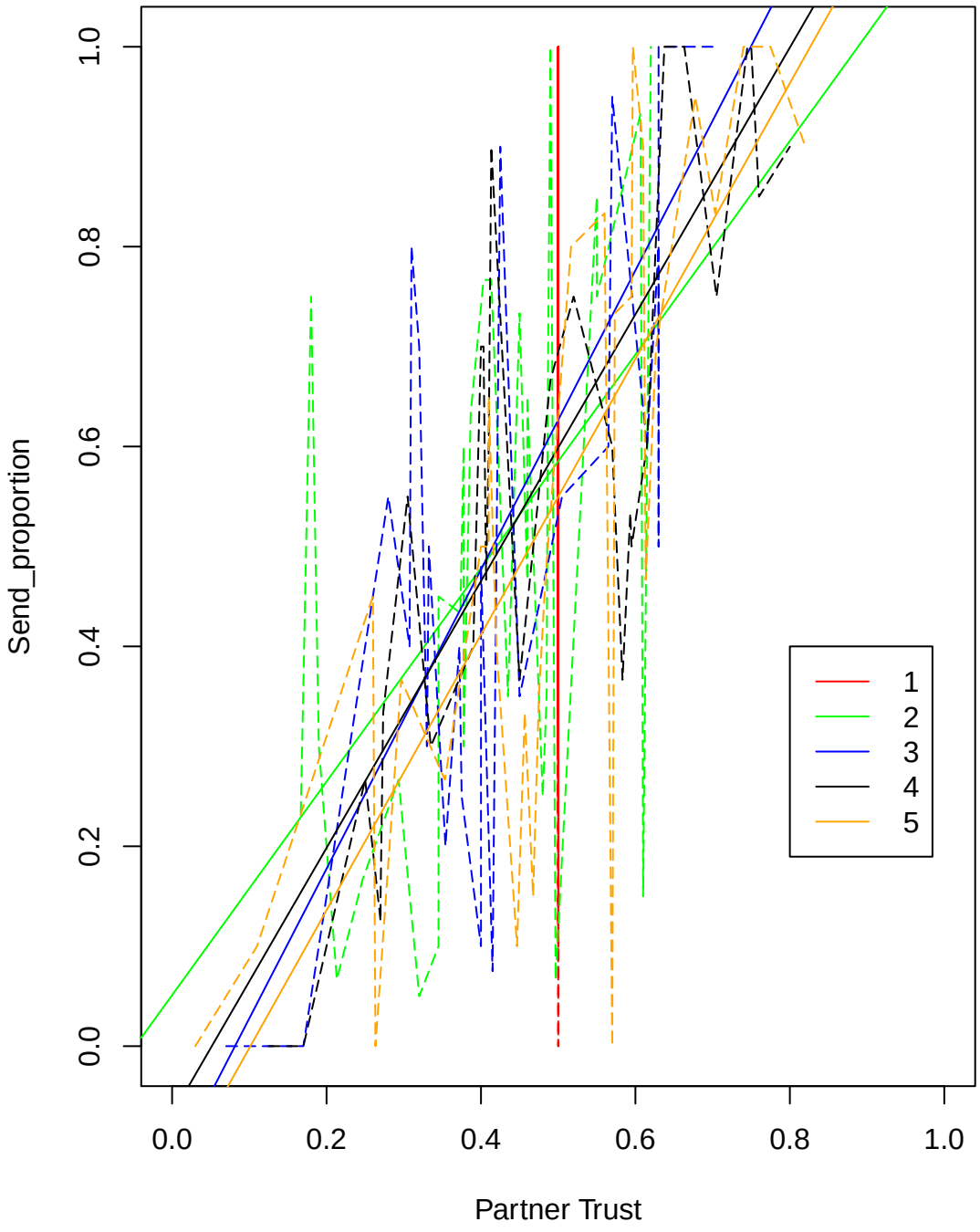
```
##
##
## \pagebreak
```


Score Game SENDER
Send_proportion
Partner Trust
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

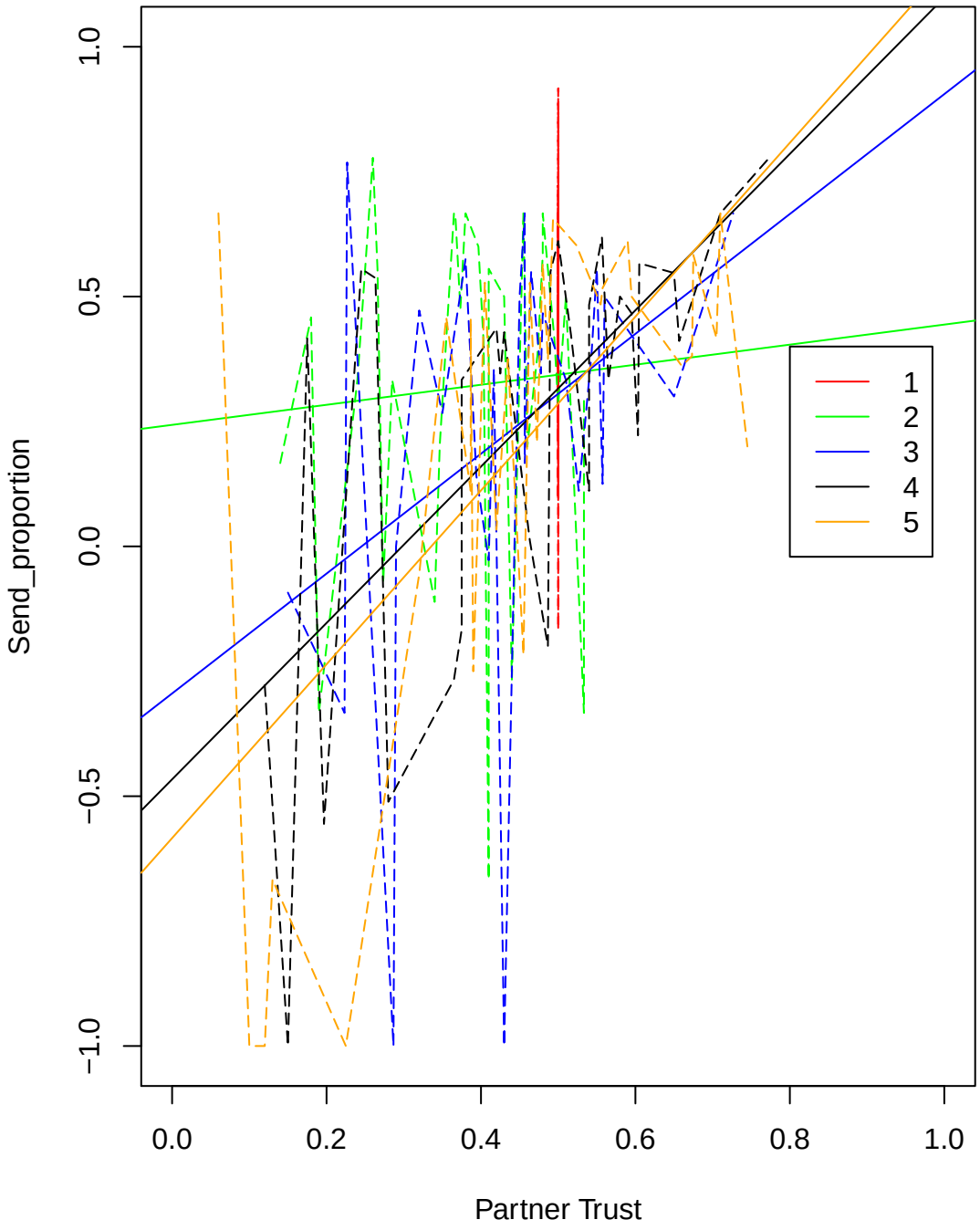
```
##
##
## \pagebreak
```


Score Game RECEIVER
Send_proportion
Partner Trust
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

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


Combine Game SENDER
Send_proportion
Partner Trust
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

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


Combine Game RECEIVER
Send_proportion
Partner Trust
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5

```
##
##
## \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 consitency 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 consitency between groups. From the above analysis, we can see that Group 3 is somehow strange, so we want to analyze without them.

Temorary 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

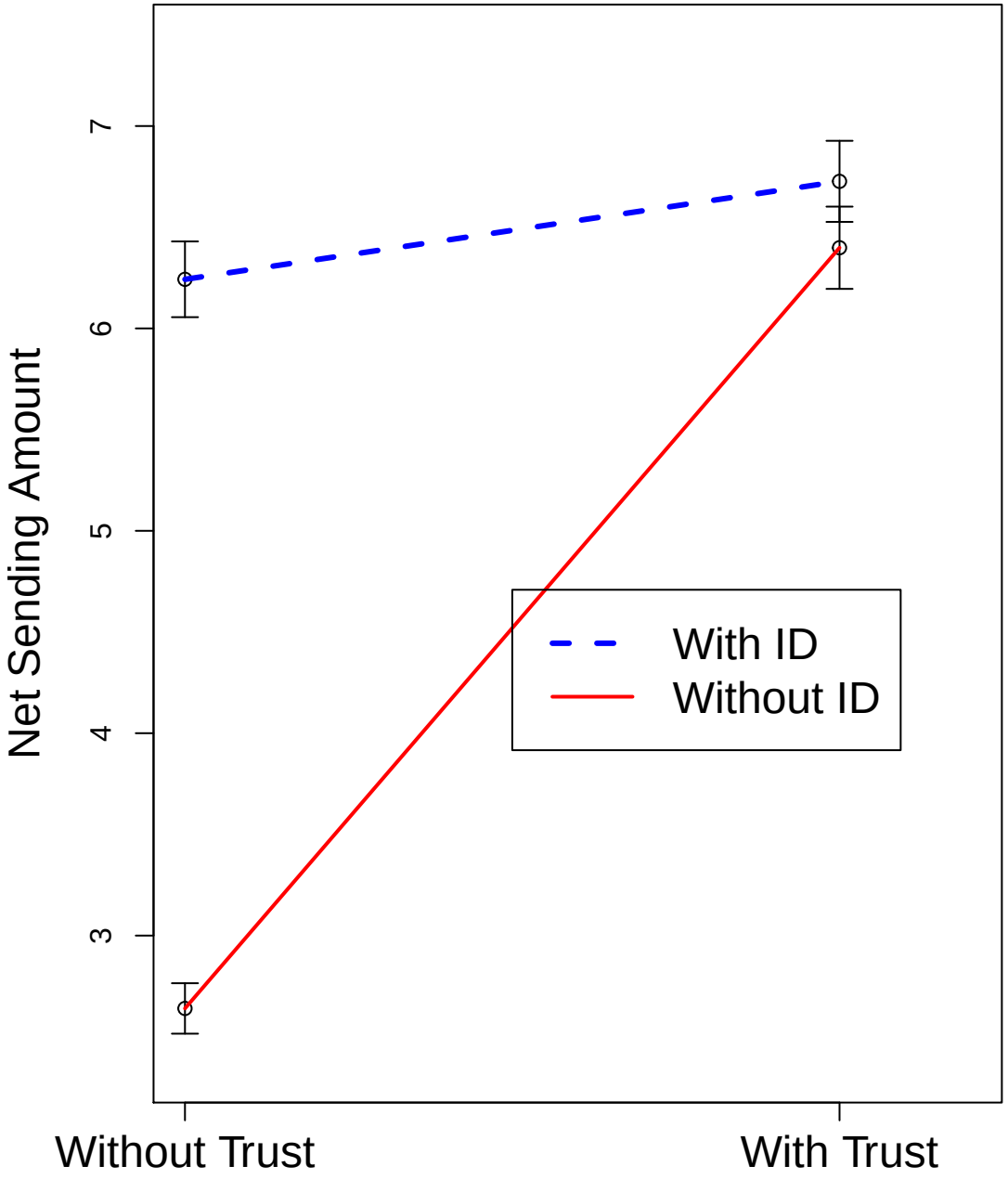
Analyze on game and group and Tukey test

We analyze the difference between each group for each game

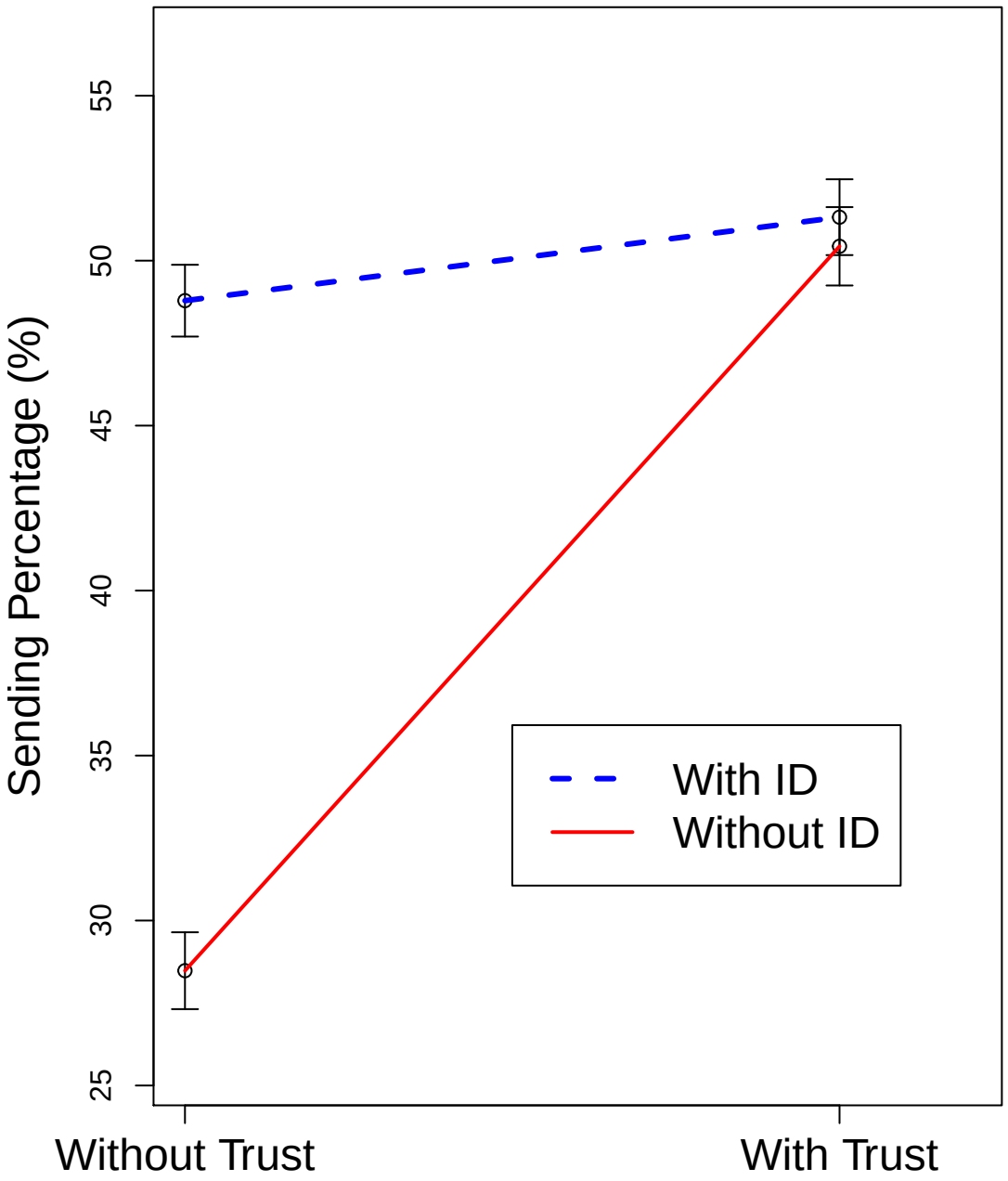
Plotting data

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

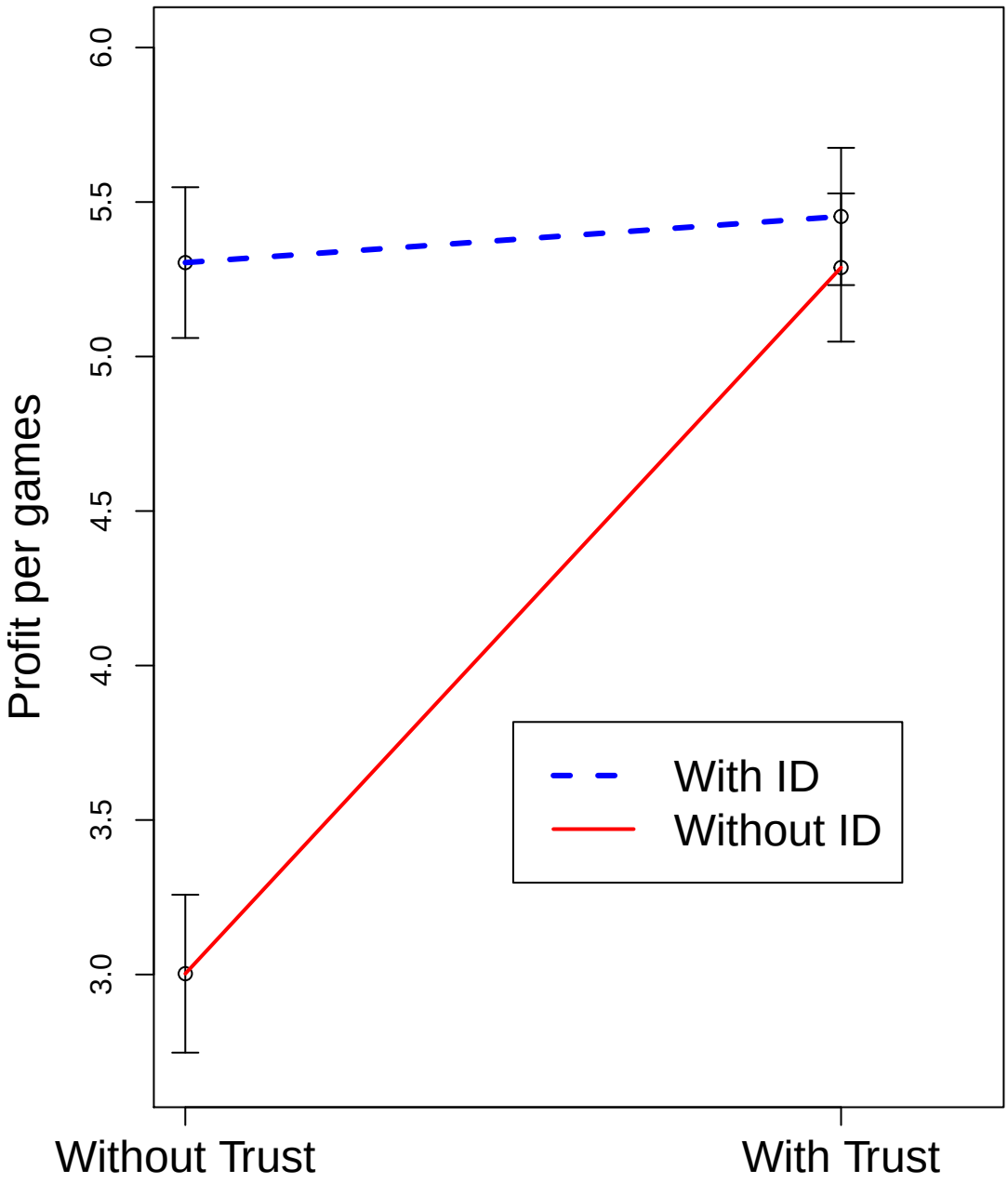
```
plotGameMetrics(simple_games$Contribution, id_games$Contribution,
    score_games$Contribution, combine_games$Contribution,
    metric_name = "Net Sending Amount")
```




```
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 (%)")
```

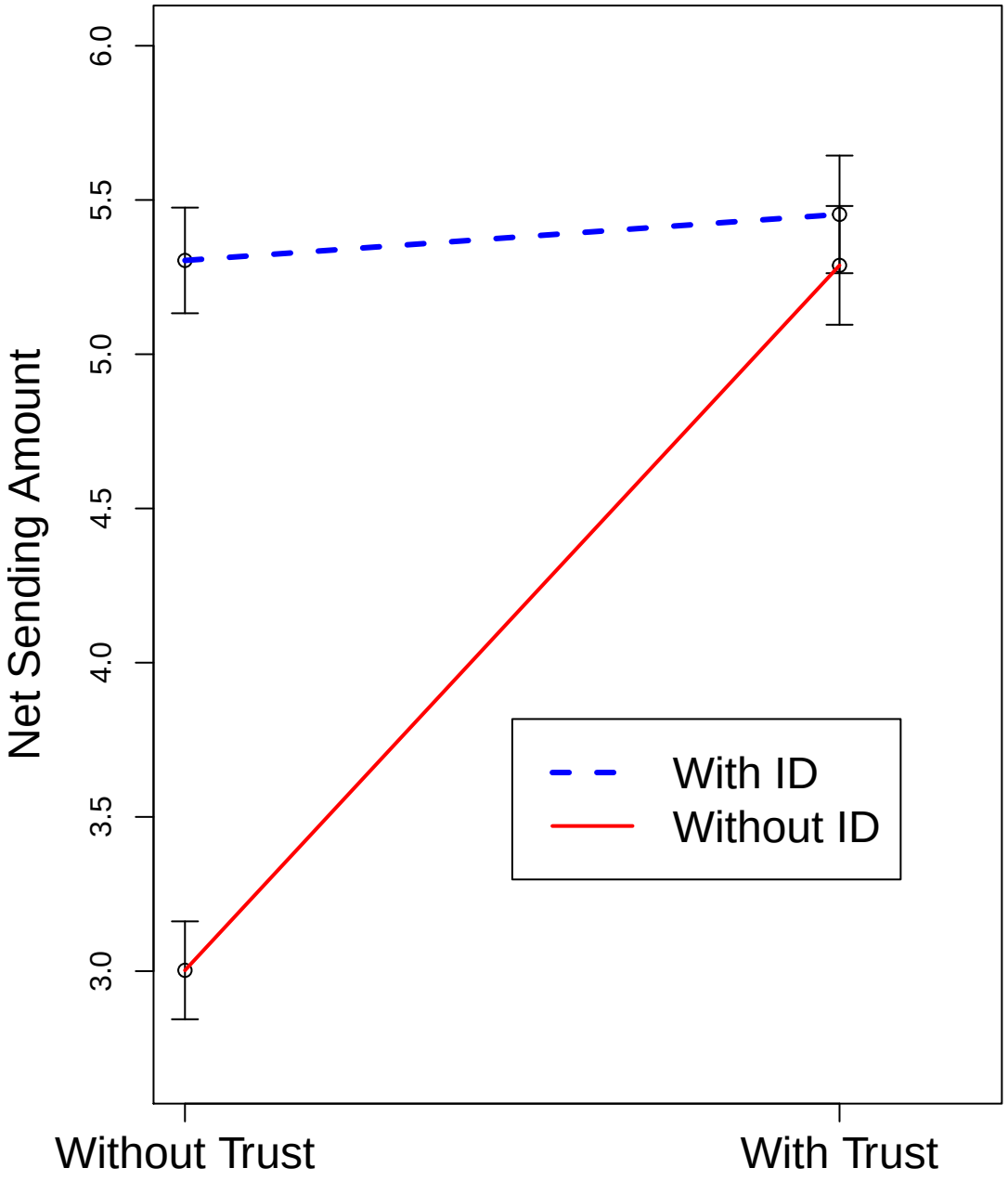


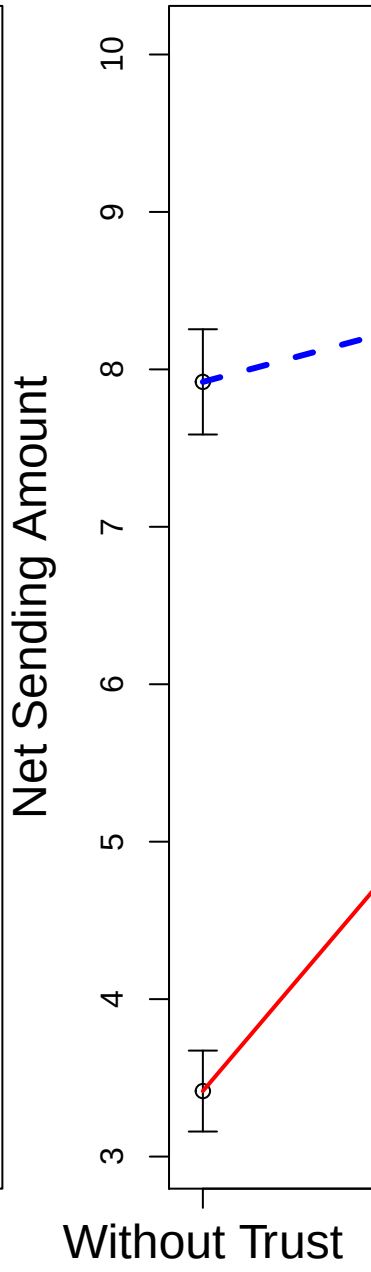

```
plotGameMetrics(simple_games$CurrGameProfit, id_games$CurrGameProfit,
    score_games$CurrGameProfit, combine_games$CurrGameProfit,
    metric_name = "Profit per games")
```

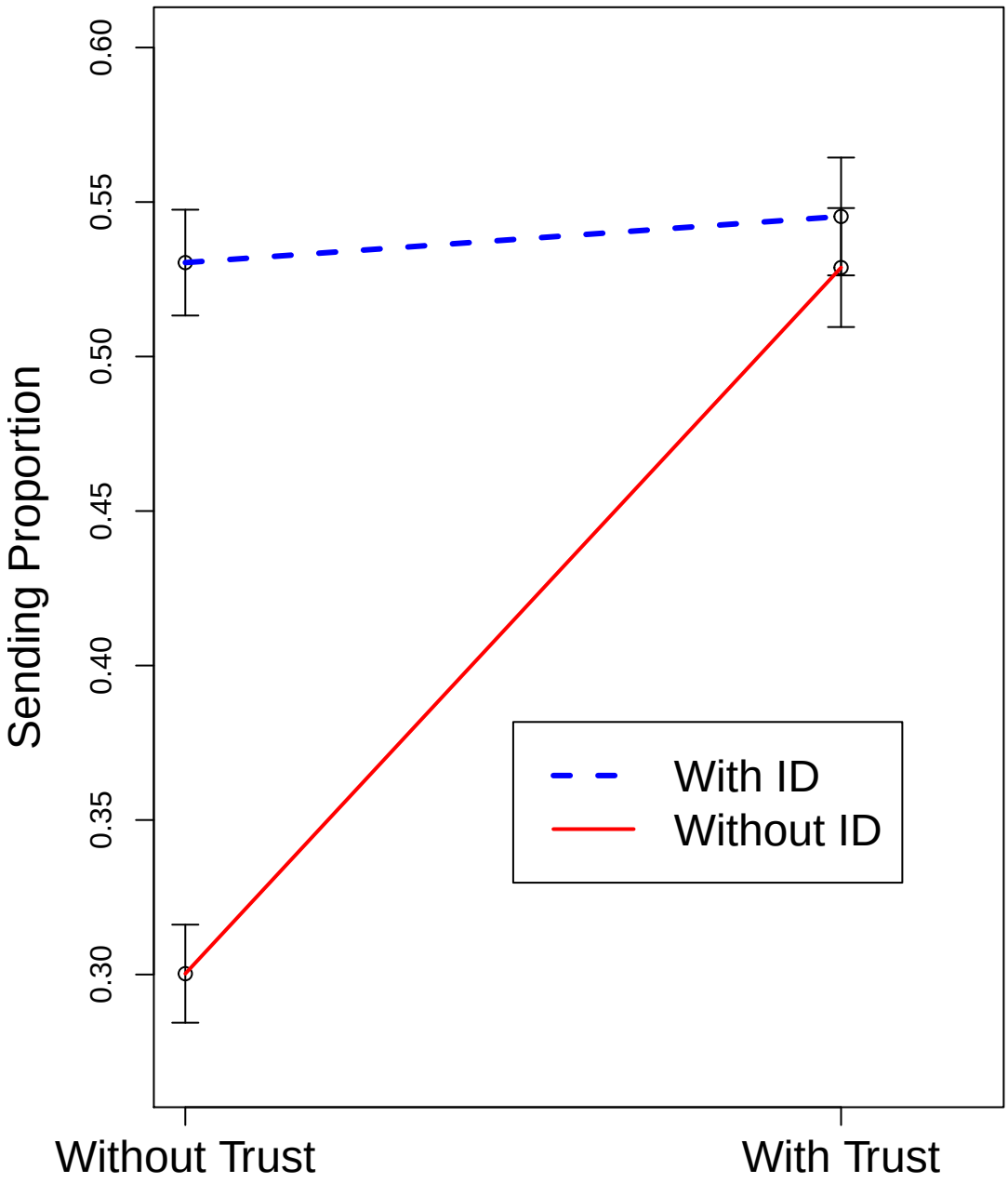

Profit per games
6.0
5.5
5.0
4.5
4.0
3.5
3.0
With ID
Without ID
Without Trust
With Trust

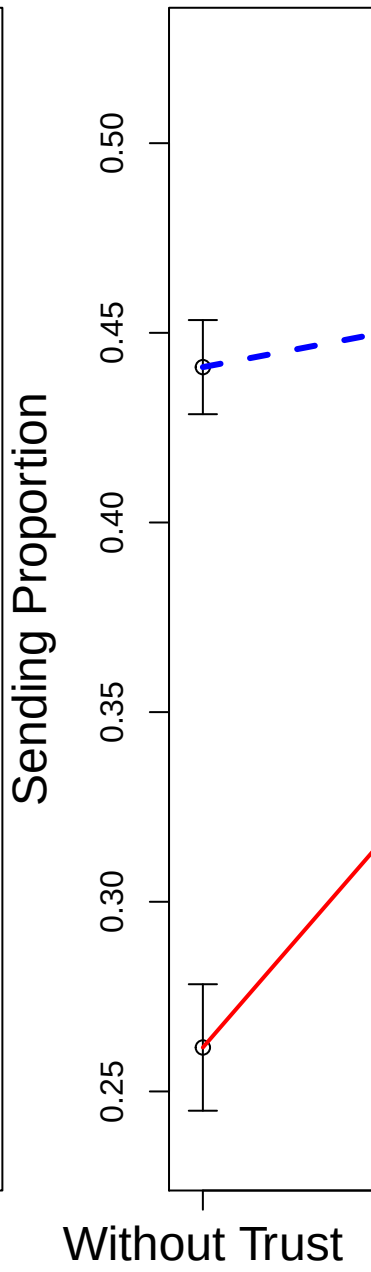
Date: Tue 7-Jun-2017

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






Sending Proportion
0.60
0.55
0.50
0.45
0.40
0.35
0.30
With ID
Without ID
Without Trust
With Trust


Sending Proportion
0.50
0.45
0.40
0.35
0.30
0.25
Without Trust

Comparing data

Giangiacomo

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")

# id: participant id BS: the session was performed
# in Brescia CN: the session was performed in Cuneo
# type: 1 = player A; 2 = player B (role definition
# in the article) daAaB: amount invested daBaA:
# amount returned actualDaAaB: amount invested by
# the opponent when playing as B actualDaBaA:
# amount returned by the opponent when playing as A
# prevA: amount received last time the subject
# played as A (= last non NA actualDaBaA) prevB:
# amount received last time the subject played as B
# (= last non NA actualDaAaB)



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

data2$send_proportion <- ifelse(data2$type == 1, data2$daAaB/10,
    data2$daBaA/3/data2$actualDaAaB)
data2$send_proportion <- ifelse(data2$send_proportion >
    1, 1, data2$send_proportion)

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))
```

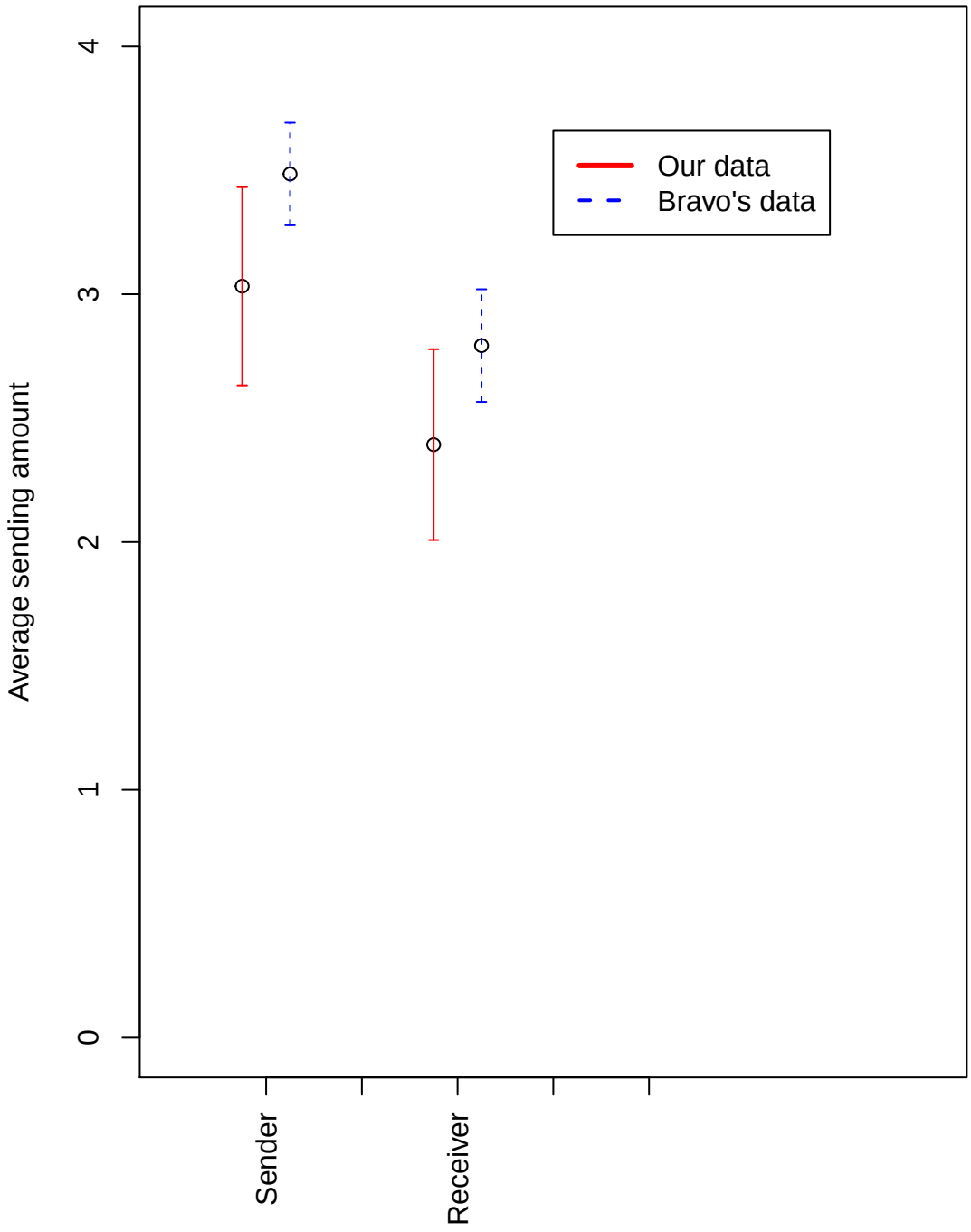
```
##
##  Welch Two Sample t-test
##
## data:  sent1$x and sent2$x
## t = -0.99144, df = 45.3, p-value = 0.3267
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.3726571  0.4669504
## sample estimates:
## mean of x mean of y
##  3.032332  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
Average sending amount
0
1
2
3
4
Sender
Receiver
Our data
Bravo's 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
```

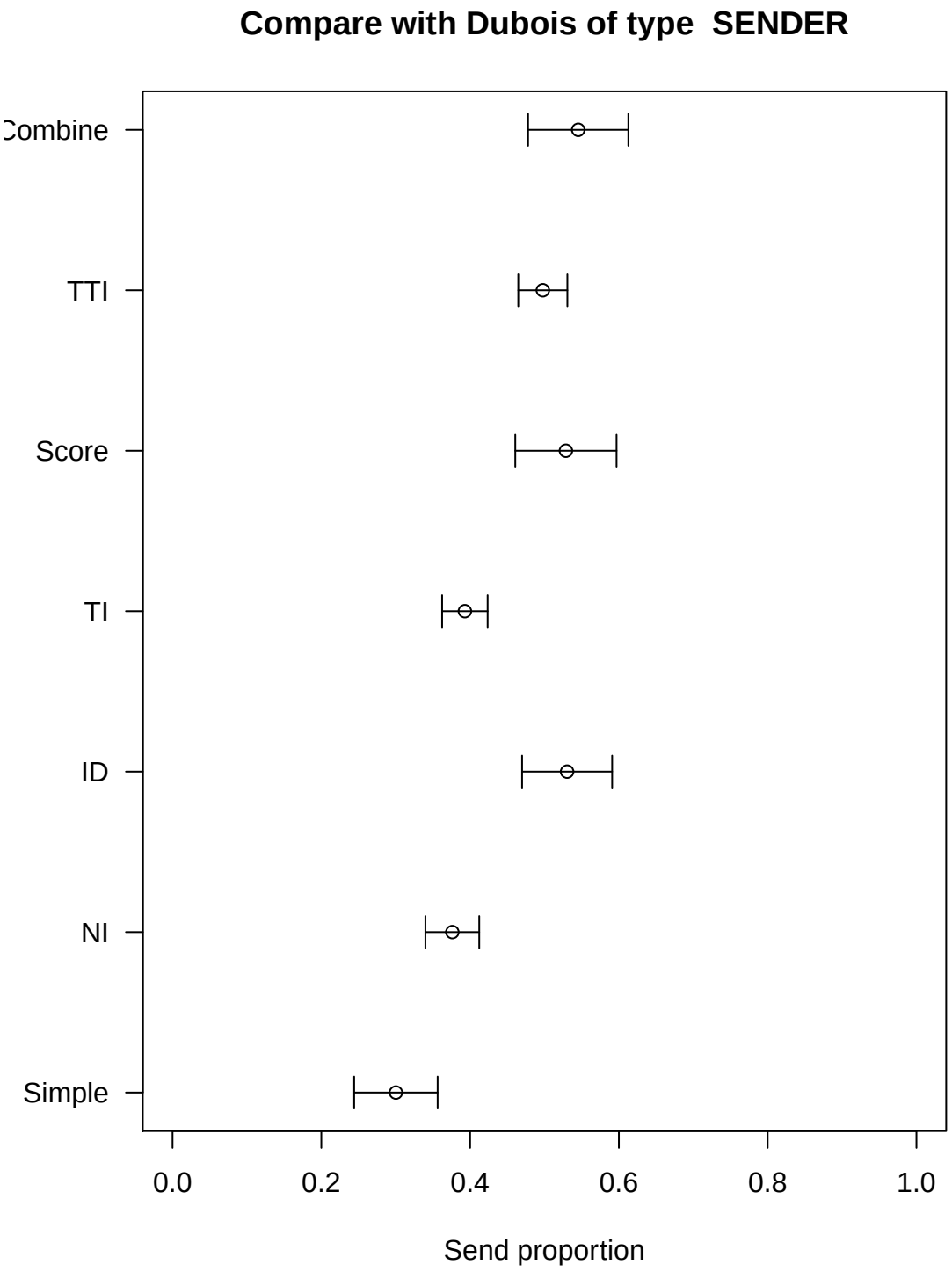
```
res2 <- lm(user_sender_payoff ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
    df2 = a$Df[2])
print(a)
```

```
## 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
```

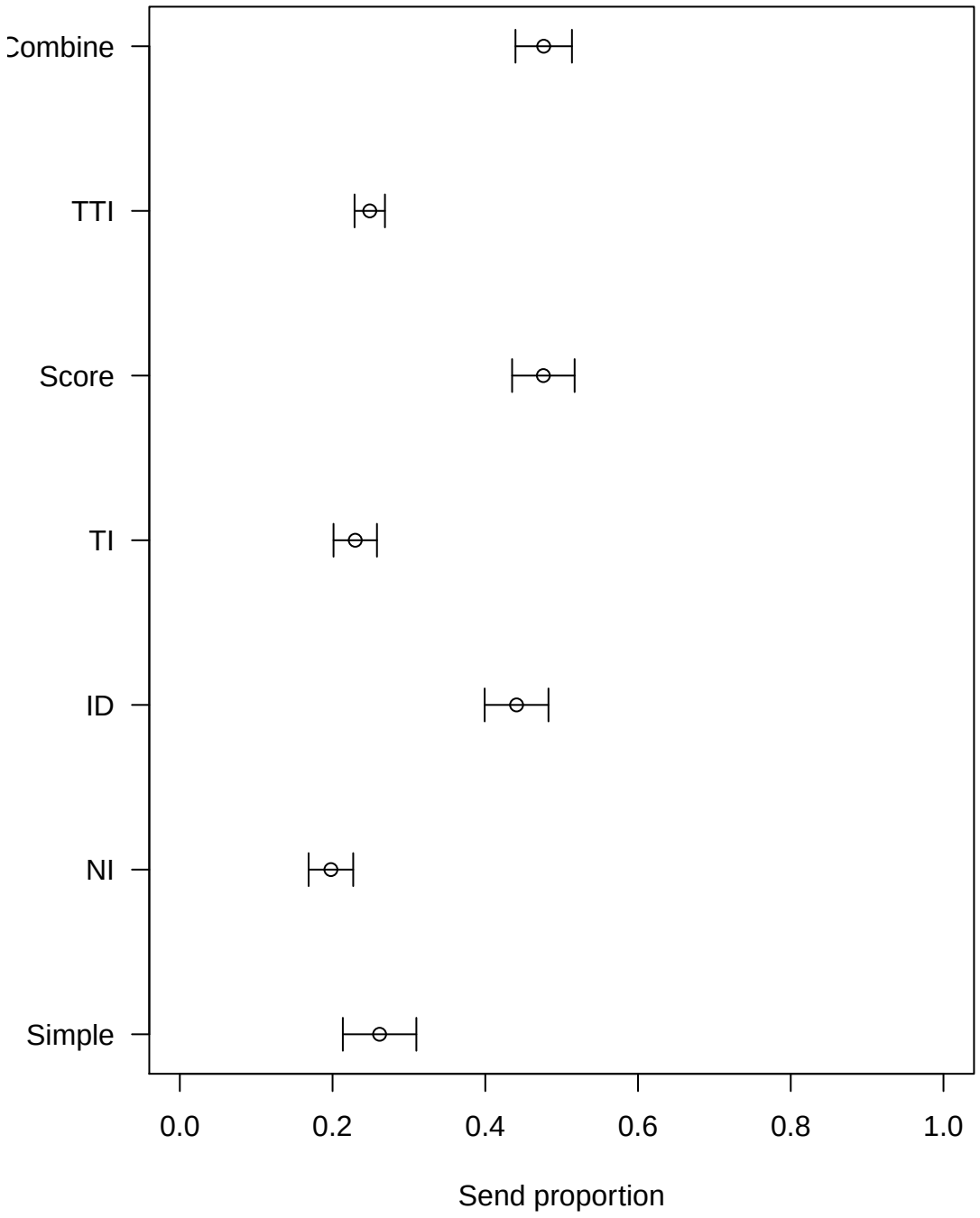
```
res2 <- lm(user_receiver_payoff ~ treatment + treatment/group)
a <- anova(res2)
a$`F value`[1] <- (a$`Sum Sq`[1]/a$Df[1])/(a$`Sum Sq`[2]/a$Df[2])
a$`Pr(>F)`[1] <- 1 - pf(a$`F value`[1], df1 = a$Df[1],
    df2 = a$Df[2])
print(a)
```

```
## 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




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




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

```
# Plot send proportion for three datasets Date: Thu
# 12-Jun-2017
require(plotrix)
means <- c()
std_errors <- c()
```

```r
for (type in 0:1) {

    data_dubois <- as.numeric()
    if (type == 0) {
        data_dubois <- user_sent[1:36]/10
    } else if (type == 1) {
        data_dubois <- user_reciprocity[1:36]
    }
    # Our data
    our_data <- simple_games[simple_games$Type == type &
        simple_games$my_send_proportional >= 0, ]
    our_send_proportion <- aggregate(our_data$my_send_proportional,
        list(our_data$Subject, our_data$Date), mean)$x

    means <- c(means, mean(our_send_proportion))
    std_errors <- c(std_errors, pop.sd(simple_games[simple_games$Type ==
        type & simple_games$my_send_proportional >=
        0, ]$my_send_proportional)/sqrt(30))

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

    # Bravo
    bravo <- data2
    bravo_send_proportion <- aggregate(bravo[bravo$type ==
        type + 1 & !is.na(bravo$send_proportion), ]$send_proportion,
        list(bravo[bravo$type == type + 1 & !is.na(bravo$send_proportion),
            ]$id), mean)$x
    means <- c(means, mean(bravo_send_proportion))
    std_errors <- c(std_errors, pop.sd(bravo_send_proportion)/sqrt(108))

    print(paste("t-test between send proportions:",
        Type_names[type + 1]))

    print("With Dubois")
    print(t.test(our_send_proportion, data_dubois))

    print("With Bravo")
    bravo_send_proportion <- aggregate(bravo[bravo$type ==
        type + 1 & !is.na(bravo$send_proportion), ]$send_proportion,
        list(bravo[bravo$type == type + 1 & !is.na(bravo$send_proportion),
            ]$id), mean)$x
    print(t.test(our_send_proportion, bravo_send_proportion))

}
```

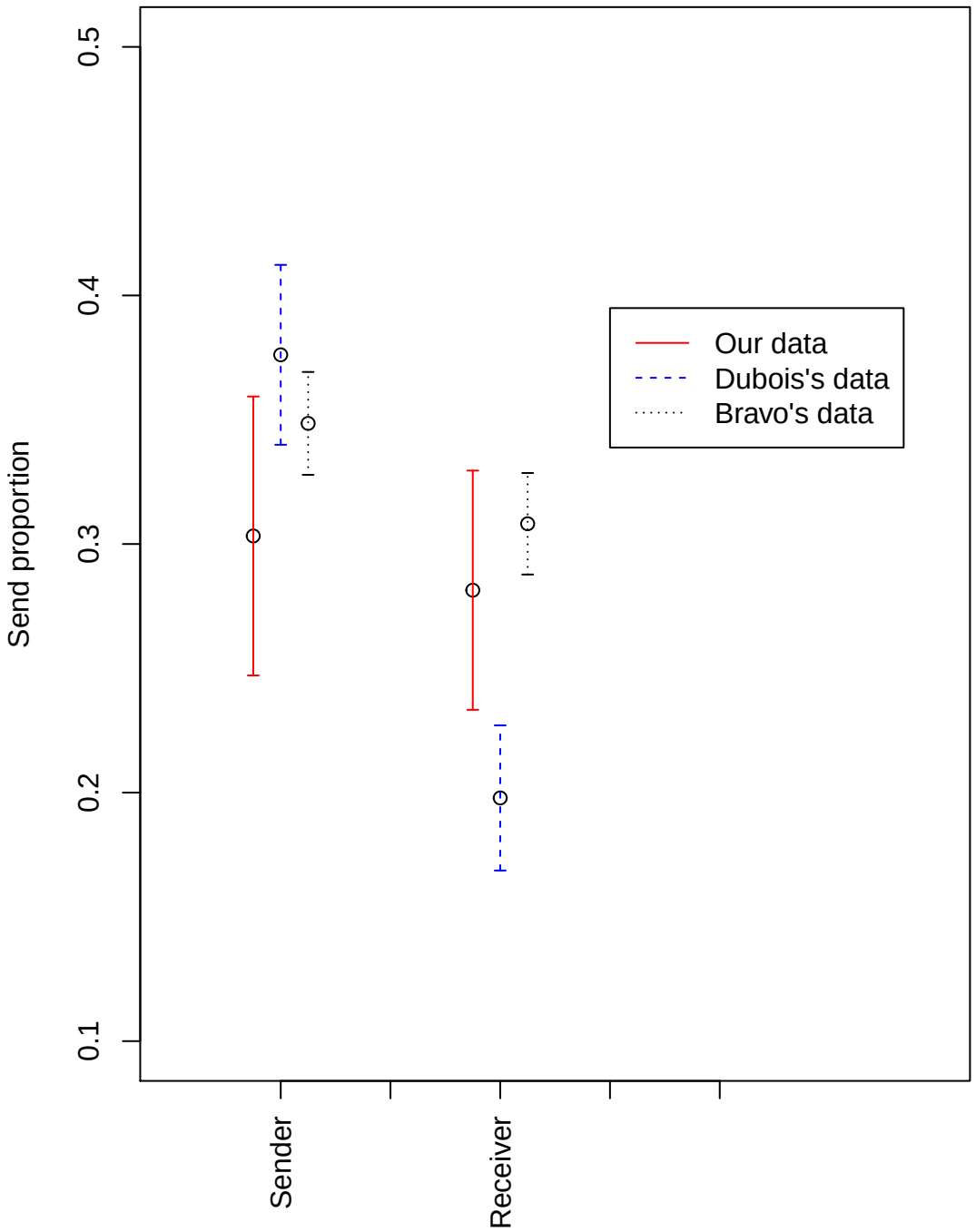
```
## [1] "t-test between send proportions: SENDER"
## [1] "With Dubois"
##
##  Welch Two Sample t-test
##
## data:  our_send_proportion and data_dubois
## t = -1.3304, df = 61.606, p-value = 0.1883
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1823901  0.0366342
## sample estimates:
## mean of x mean of y
## 0.3032332 0.3761111
##
## [1] "With Bravo"
##
##  Welch Two Sample t-test
##
## data:  our_send_proportion and bravo_send_proportion
## 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:
##  -0.13726571  0.04669504
## sample estimates:
## mean of x mean of y
## 0.3032332 0.3485185
##
## [1] "t-test between send proportions: RECEIVER"
## [1] "With Dubois"
##
##  Welch Two Sample t-test
##
## data:  our_send_proportion and data_dubois
## t = 1.6863, df = 55.878, p-value = 0.09732
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01571286  0.18285528
## sample estimates:
## mean of x mean of y
## 0.2814123 0.1978410
##
## [1] "With Bravo"
##
##  Welch Two Sample t-test
##
## data:  our_send_proportion and bravo_send_proportion
## t = -0.5979, df = 45.638, p-value = 0.5529
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11677459  0.06329791
## sample estimates:
## mean of x mean of y
## 0.2814123 0.3081506
```

```
xs = c(0.75, 1, 1.25, 2.75, 3, 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 = "Send proportion", main = "",
    xaxt = "n", xlim = c(0, 7), ylim = c(0.1, 0.5),
```

```
    xlab = "")
axis(1, at = 1:5, labels = c("Sender", "", "Receiver",
    "", ""), las = 2)
# Our data
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")

# Dubois
segments(x0 = 1, y0 = means[2] - std_errors[2], x1 = 1,
    y1 = means[2] + std_errors[2], col = "blue", lty = 2)
segments(x0 = 0.95, y0 = means[2] - std_errors[2],
    x1 = 1.05, y1 = means[2] - std_errors[2], col = "blue")
segments(x0 = 0.95, y0 = means[2] + std_errors[2],
    x1 = 1.05, y1 = means[2] + std_errors[2], col = "blue")

# Bravo
segments(x0 = 1.25, y0 = means[3] - std_errors[3],
    x1 = 1.25, y1 = means[3] + std_errors[3], col = "black",
    lty = 3)
segments(x0 = 1.2, y0 = means[3] - std_errors[3], x1 = 1.3,
    y1 = means[3] - std_errors[3], col = "black")
segments(x0 = 1.2, y0 = means[3] + std_errors[3], x1 = 1.3,
    y1 = means[3] + std_errors[3], col = "black")

# our
segments(x0 = 2.75, y0 = means[4] - std_errors[4],
    x1 = 2.75, y1 = means[4] + std_errors[4], col = "red")
segments(x0 = 2.7, y0 = means[4] - std_errors[4], x1 = 2.8,
    y1 = means[4] - std_errors[4], col = "red")
segments(x0 = 2.7, y0 = means[4] + std_errors[4], x1 = 2.8,
    y1 = means[4] + std_errors[4], col = "red")

# Dubois
segments(x0 = 3, y0 = means[5] - std_errors[5], x1 = 3,
    y1 = means[5] + std_errors[5], col = "blue", lty = 2)
segments(x0 = 2.95, y0 = means[5] - std_errors[5],
    x1 = 3.05, y1 = means[5] - std_errors[5], col = "blue")
segments(x0 = 2.95, y0 = means[5] + std_errors[5],
    x1 = 3.05, y1 = means[5] + std_errors[5], col = "blue")

# Bravo
segments(x0 = 3.25, y0 = means[6] - std_errors[6],
    x1 = 3.25, y1 = means[6] + std_errors[6], col = "black",
    lty = 3)
segments(x0 = 3.2, y0 = means[6] - std_errors[6], x1 = 3.3,
    y1 = means[6] - std_errors[6], col = "black")
segments(x0 = 3.2, y0 = means[6] + std_errors[6], x1 = 3.3,
    y1 = means[6] + std_errors[6], col = "black")
```

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

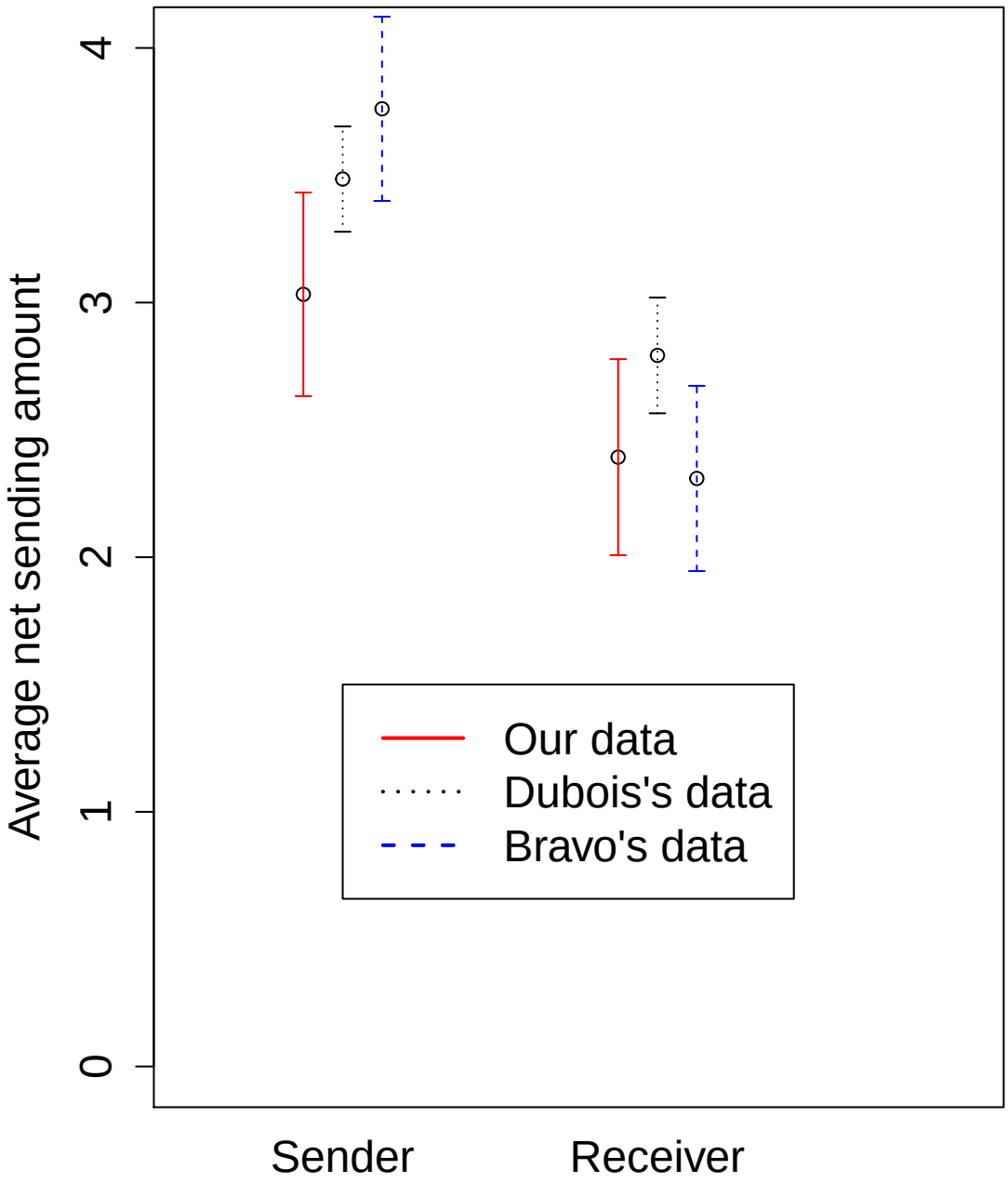


Plot net sending amount of three datasets in one figure

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

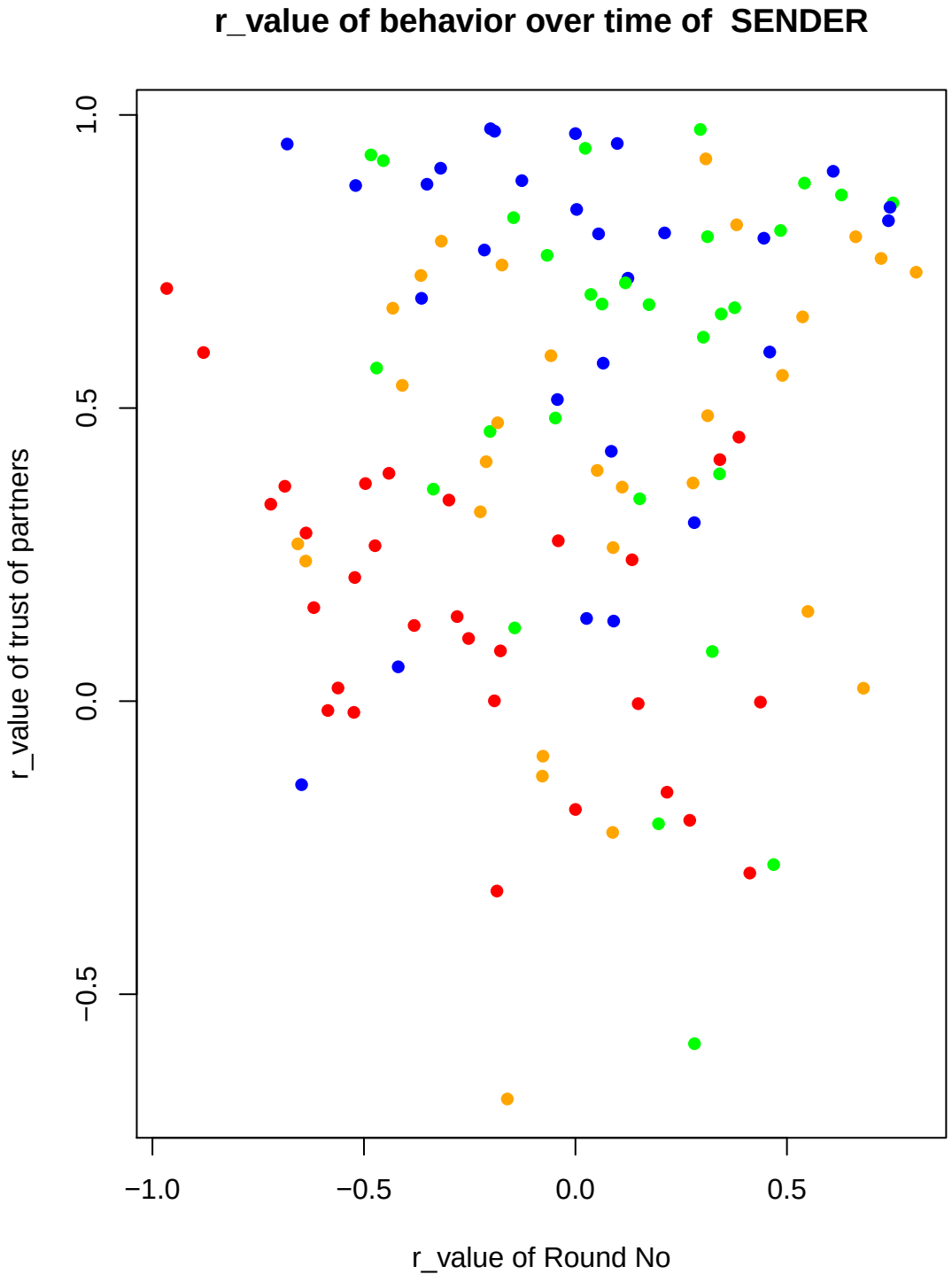
```
##
##  Welch Two Sample t-test
```

```
##
## 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 sent3/10
## t = 5.4993, df = 29.515, p-value = 5.993e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.358628 2.965649
## sample estimates:
## mean of x mean of y
## 2.3930642 0.2309259
```


Average net sending amount
0
1
2
3
4
Sender
Receiver
Our data
Dubois's data
Bravo's data

Behavior over time

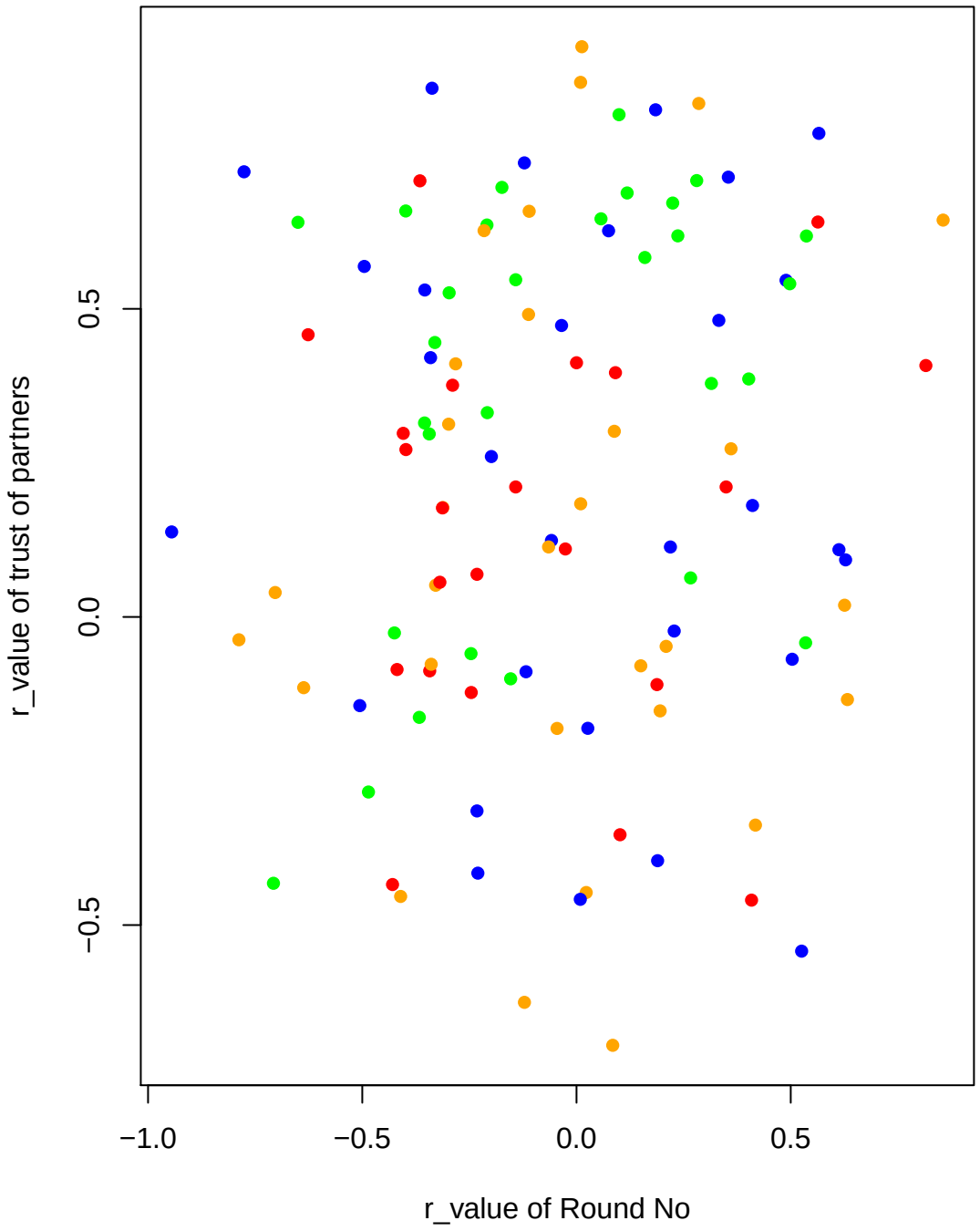
After reading the questionnaire, there is a hypothesis that, people send less in the end of game (i.e, people learn the length of the game and adapt to the game). However, after running the regression test between sending behavior and period, there is no evidence to prove the hypothesis.




```
##                         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] "---****---"
```

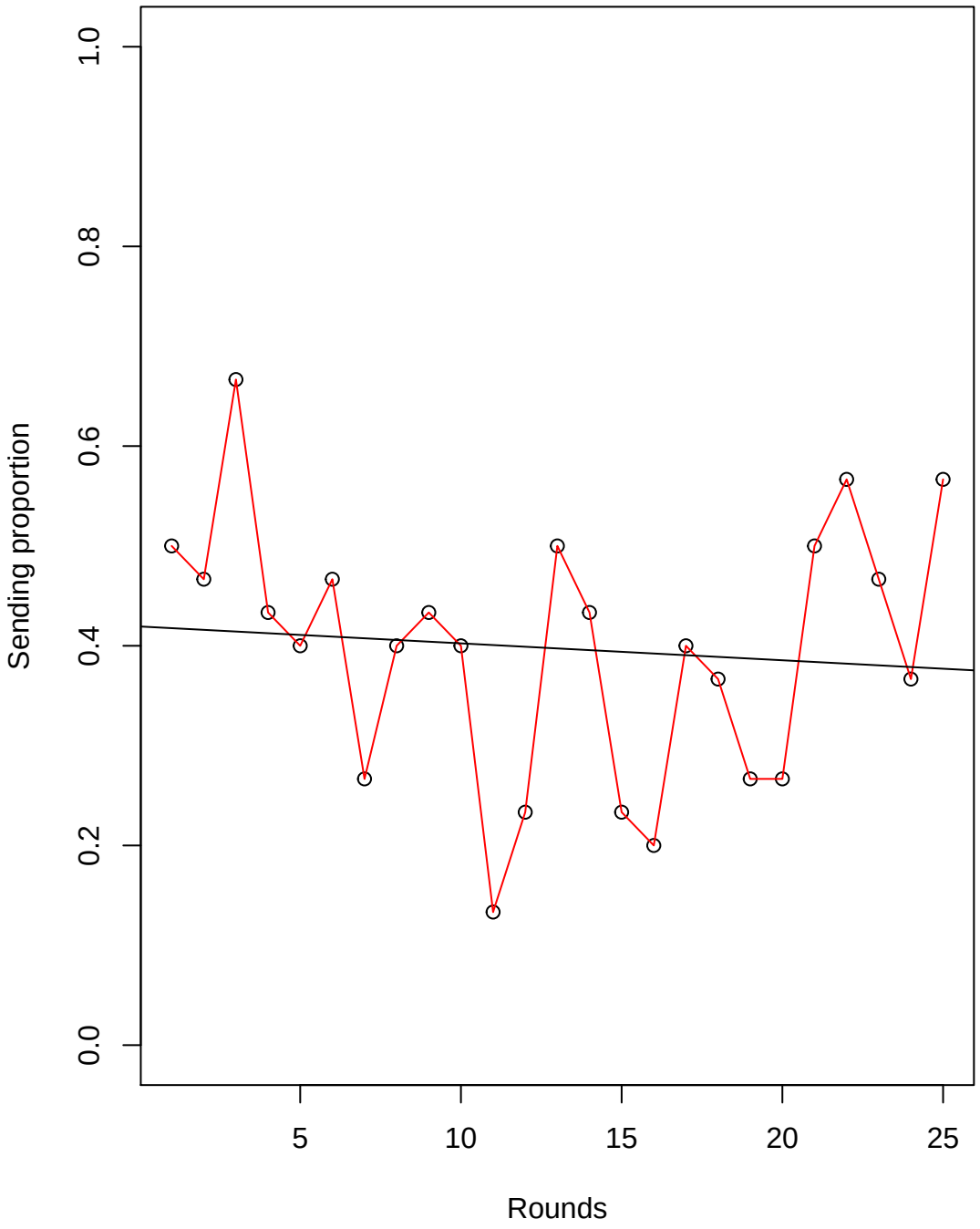
```
## [1] "Anova analysis for R value"
## [1] "-------*******--------"
## [1] "ANOVA 3-ways Analysis in wide format for type (with corrected error terms):  Sender"
##                                SS num Df Error SS den Df       F Pr(>F)
## (Intercept)                0.00368      1   1.9572     23  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
```



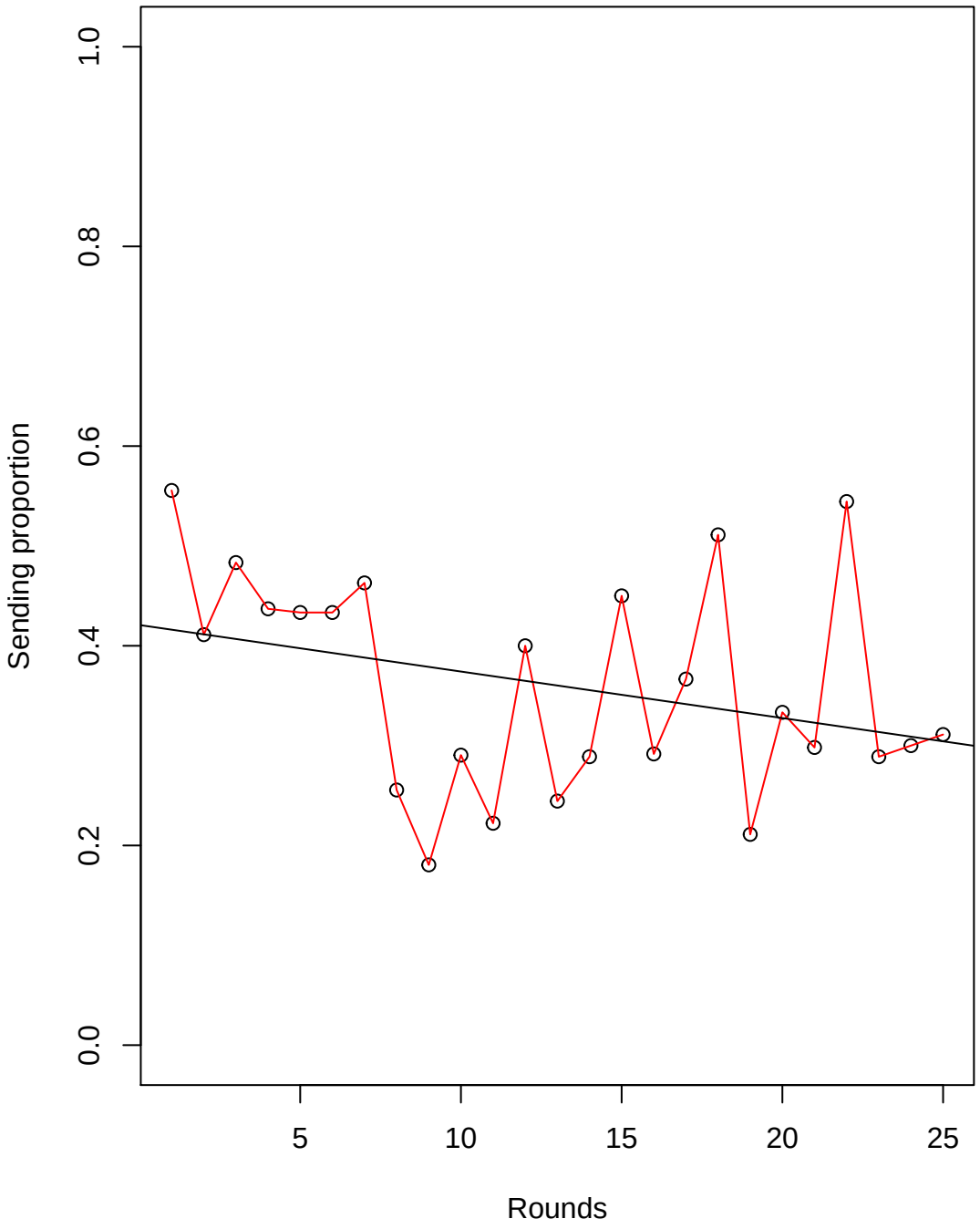

```
##                       NA      NA      NA
## 1     Slope Without trust 0.00826 0.00284
## 2     Slope    With trust 0.00595 0.00121
## 3 Intercept Without trust 0.72518 0.24914
## 4 Intercept    With trust 0.56743 0.15225
## 5   r_value Without trust 0.74394 0.14981
## 6   r_value    With trust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

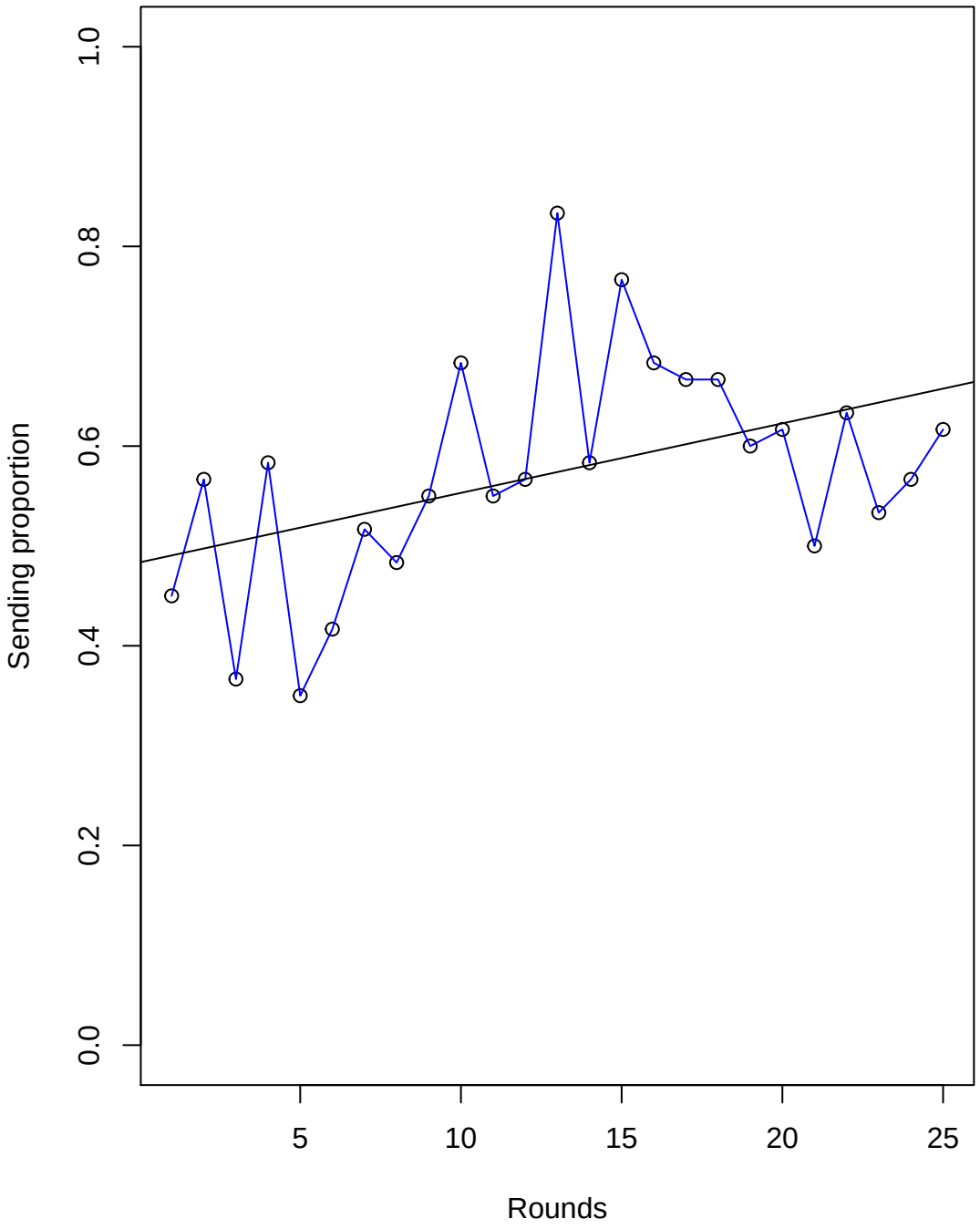
```
## GroupID:SHOW_TRUST         0.188735      4  0.55834     17 1.4366 0.2648
## SHOW_ID                    0.008131      1  0.72676     17 0.2571 0.6388
## GroupID:SHOW_ID            0.126522      4  0.72676     17 0.7399 0.5777
## SHOW_TRUST:SHOW_ID         0.002261      1  1.41863     17 0.0861 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
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

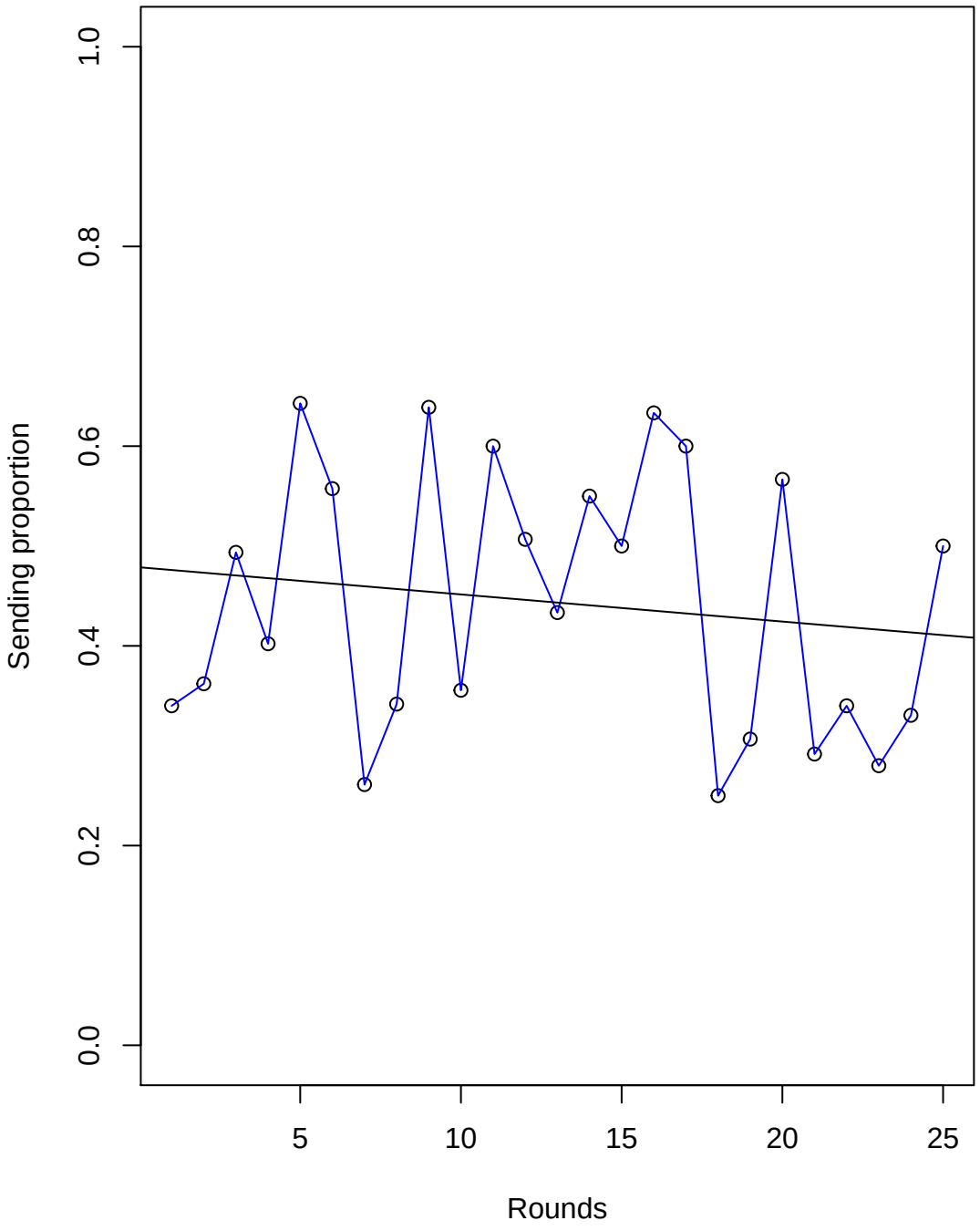
```
##
##
## \pagebreak
```


Sending behavior of Simple Games over time of: RECEIVER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

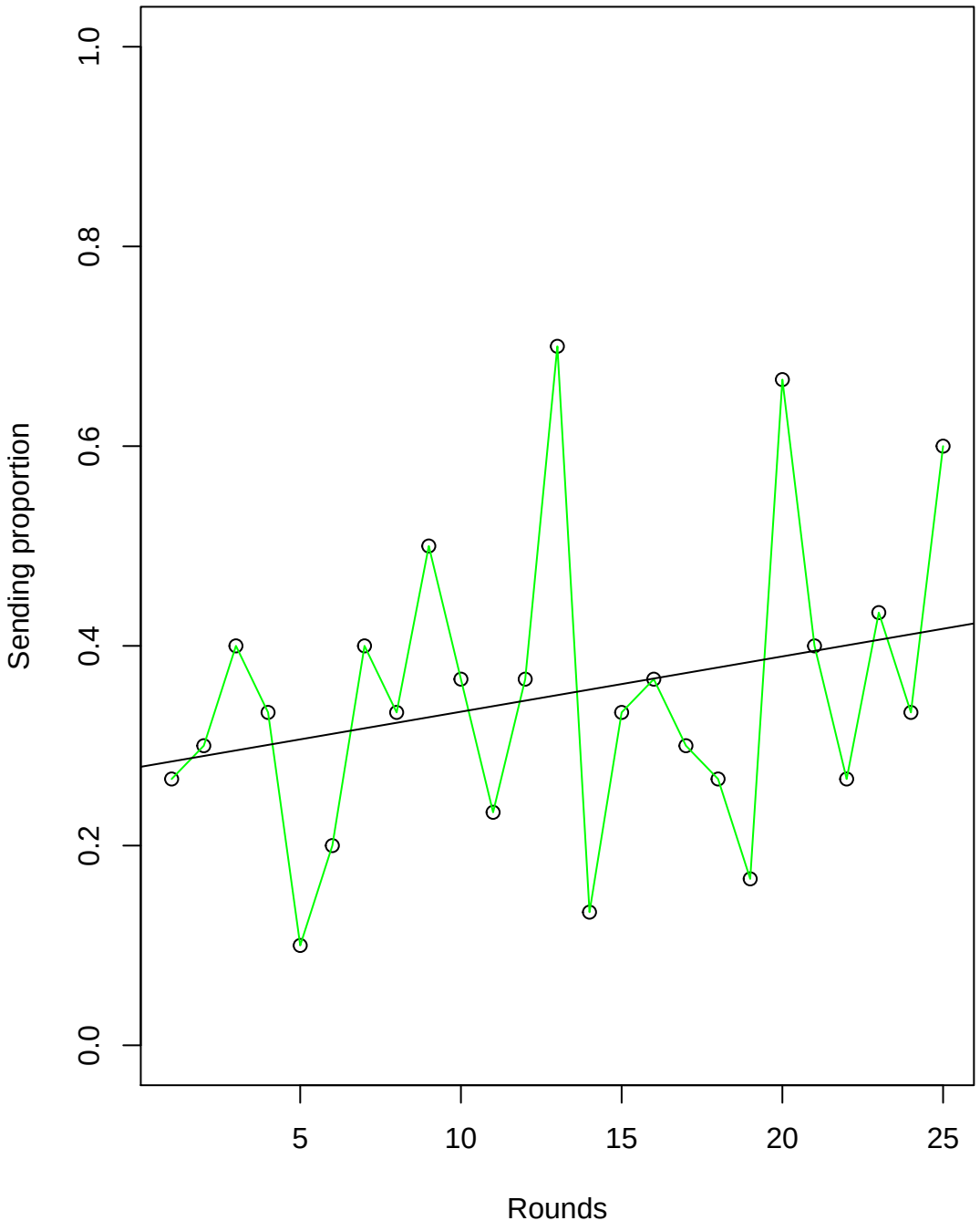
```
##
##
## \pagebreak
```


Sending behavior of ID Games over time of: SENDER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

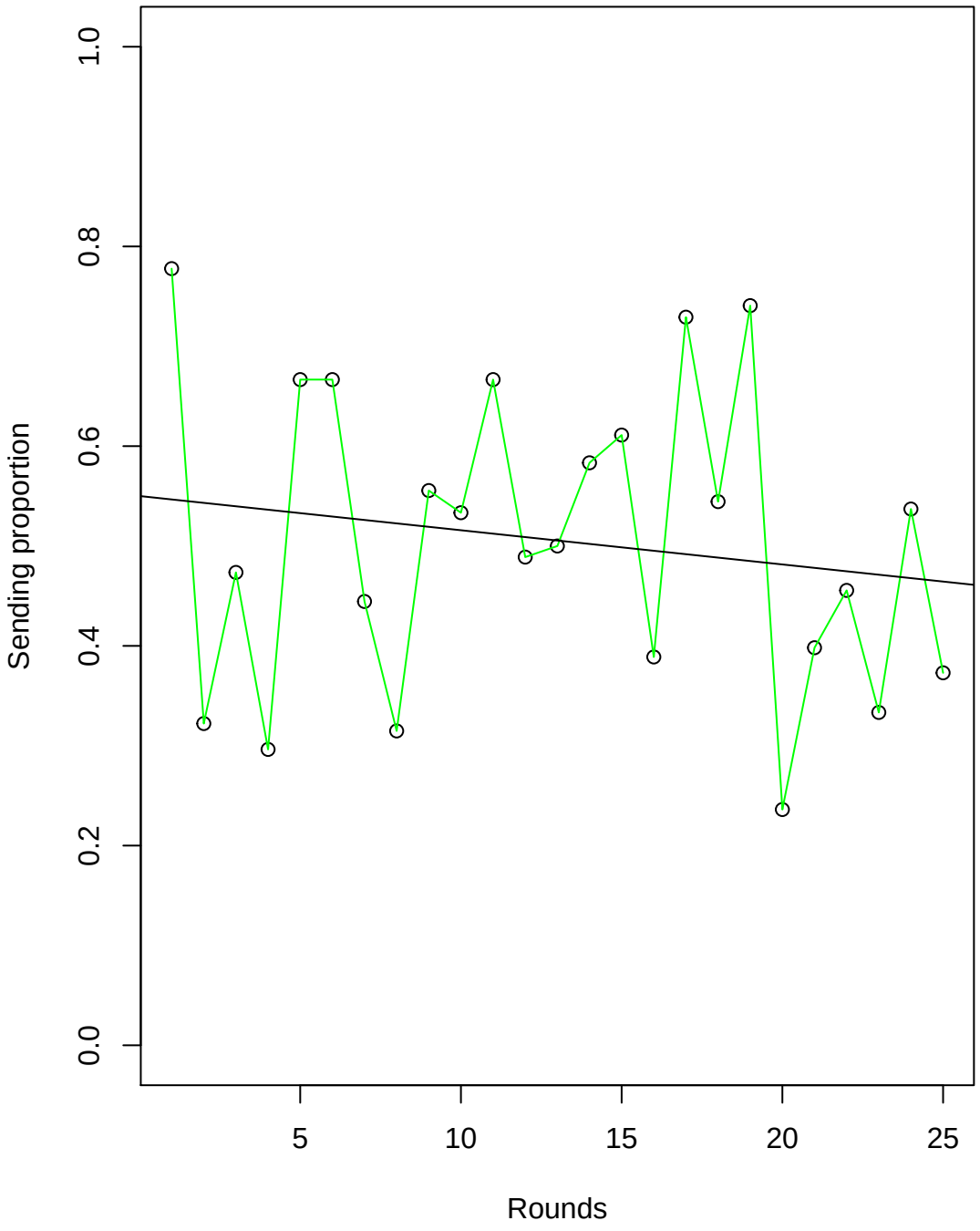
```
##
##
## \pagebreak
```


Sending behavior of ID Games over time of: RECEIVER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

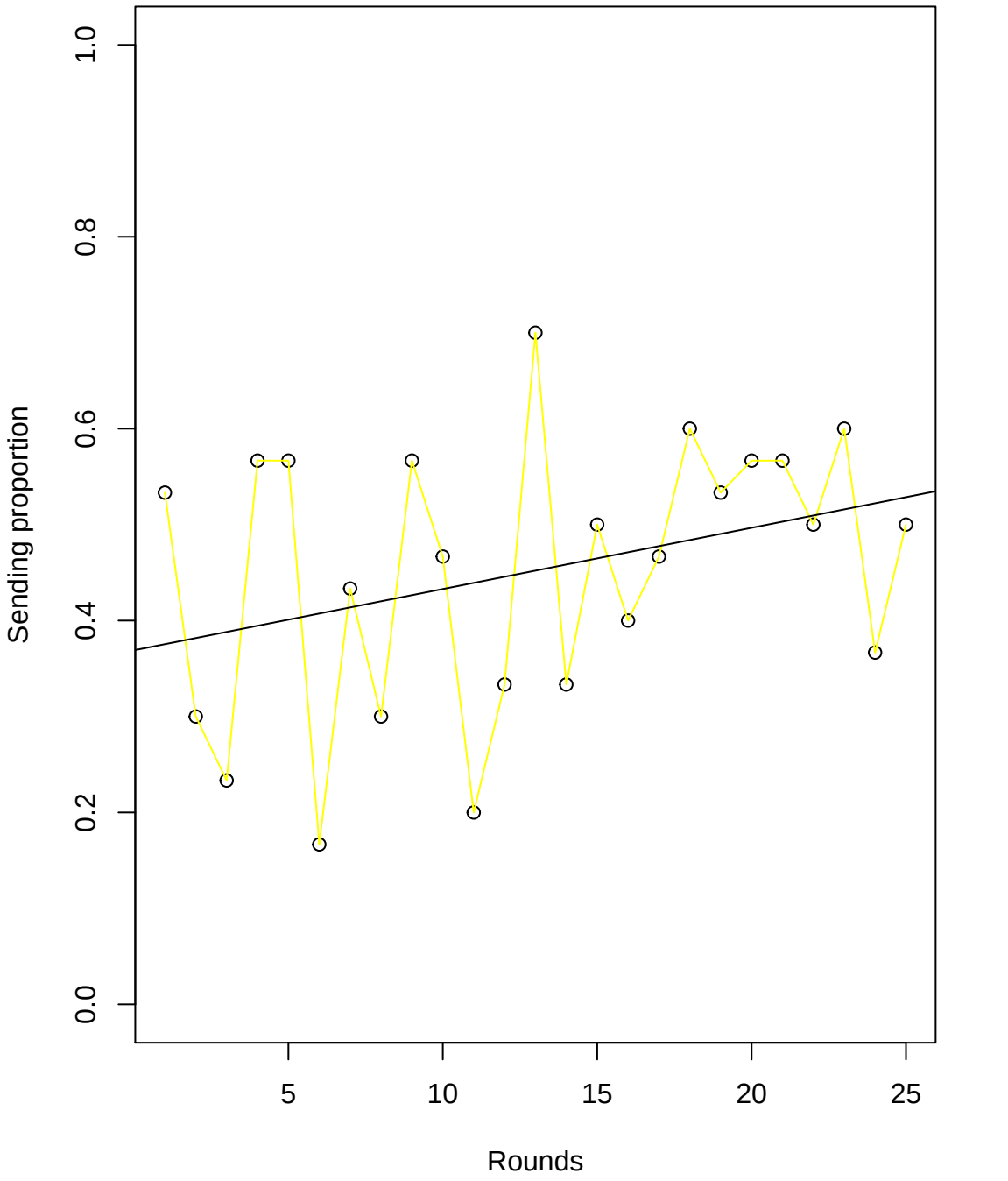
```
##
##
## \pagebreak
```


Sending behavior of Score Games over time of: SENDER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

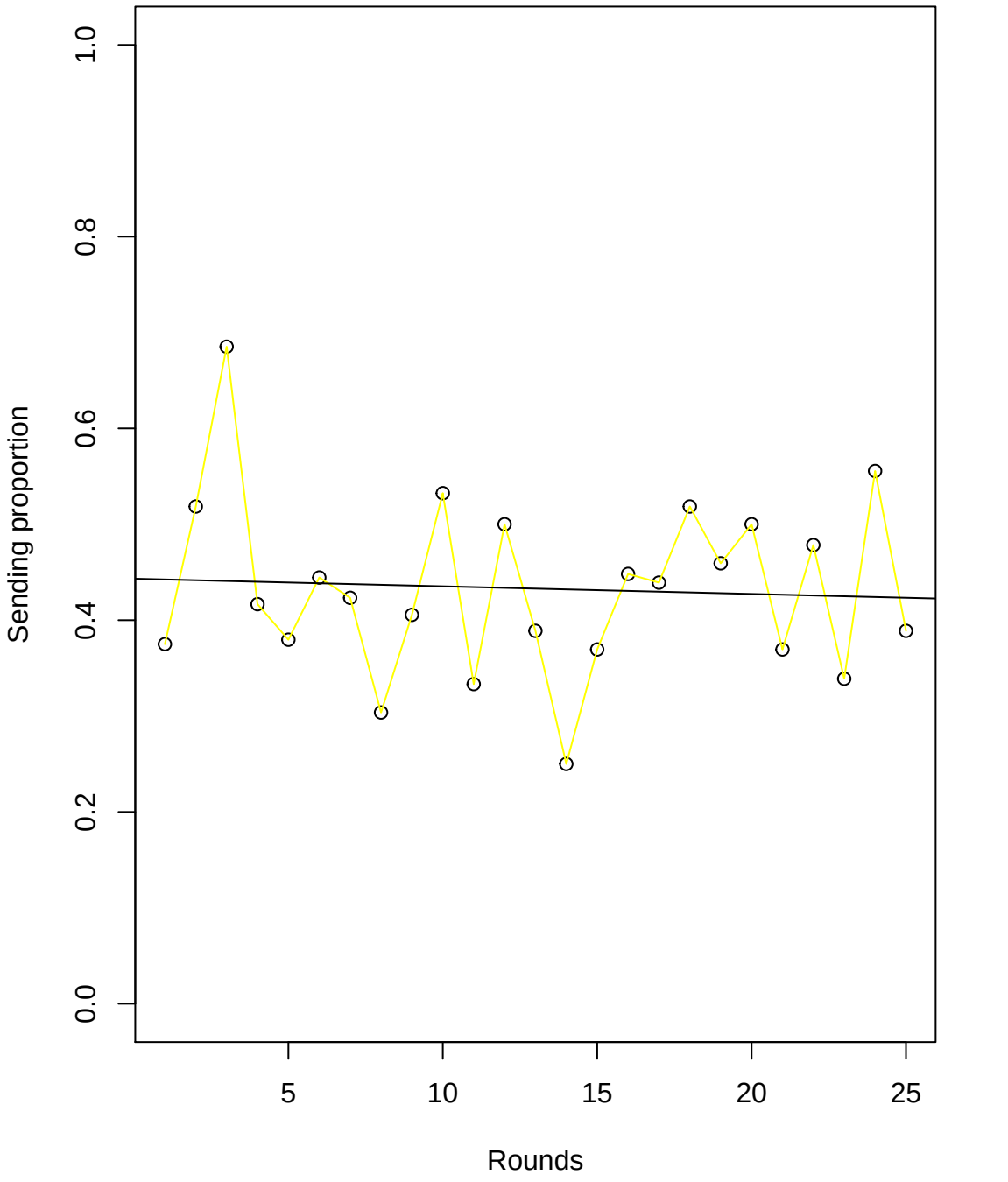
```
##
##
## \pagebreak
```


Sending behavior of Score Games over time of: RECEIVER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

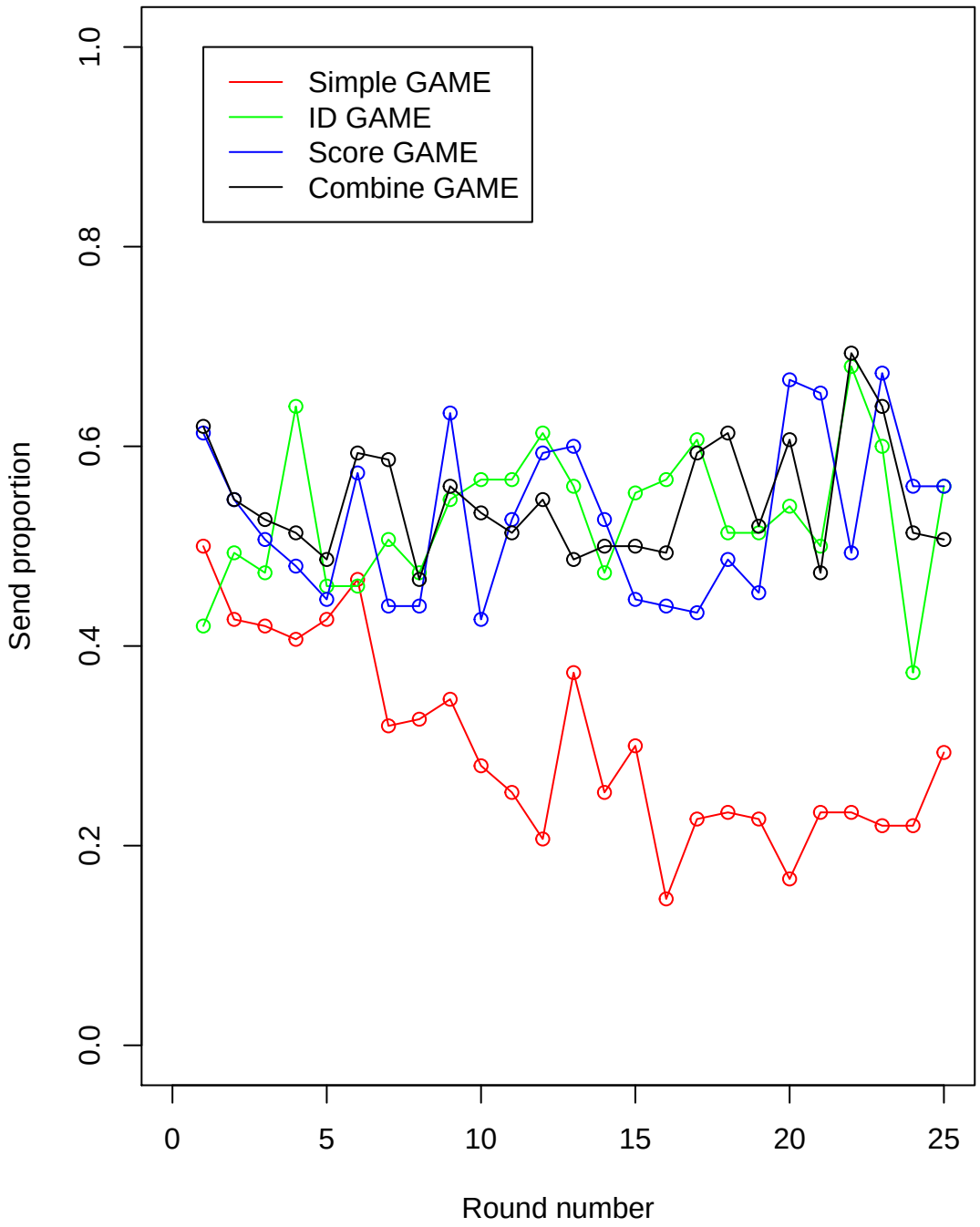
```
##
##
## \pagebreak
```


Sending behavior of Combine Games over time of: SENDER
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

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


Sending behavior of Combine Games over time of: RECEIVEI
Sending proportion
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
Rounds

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

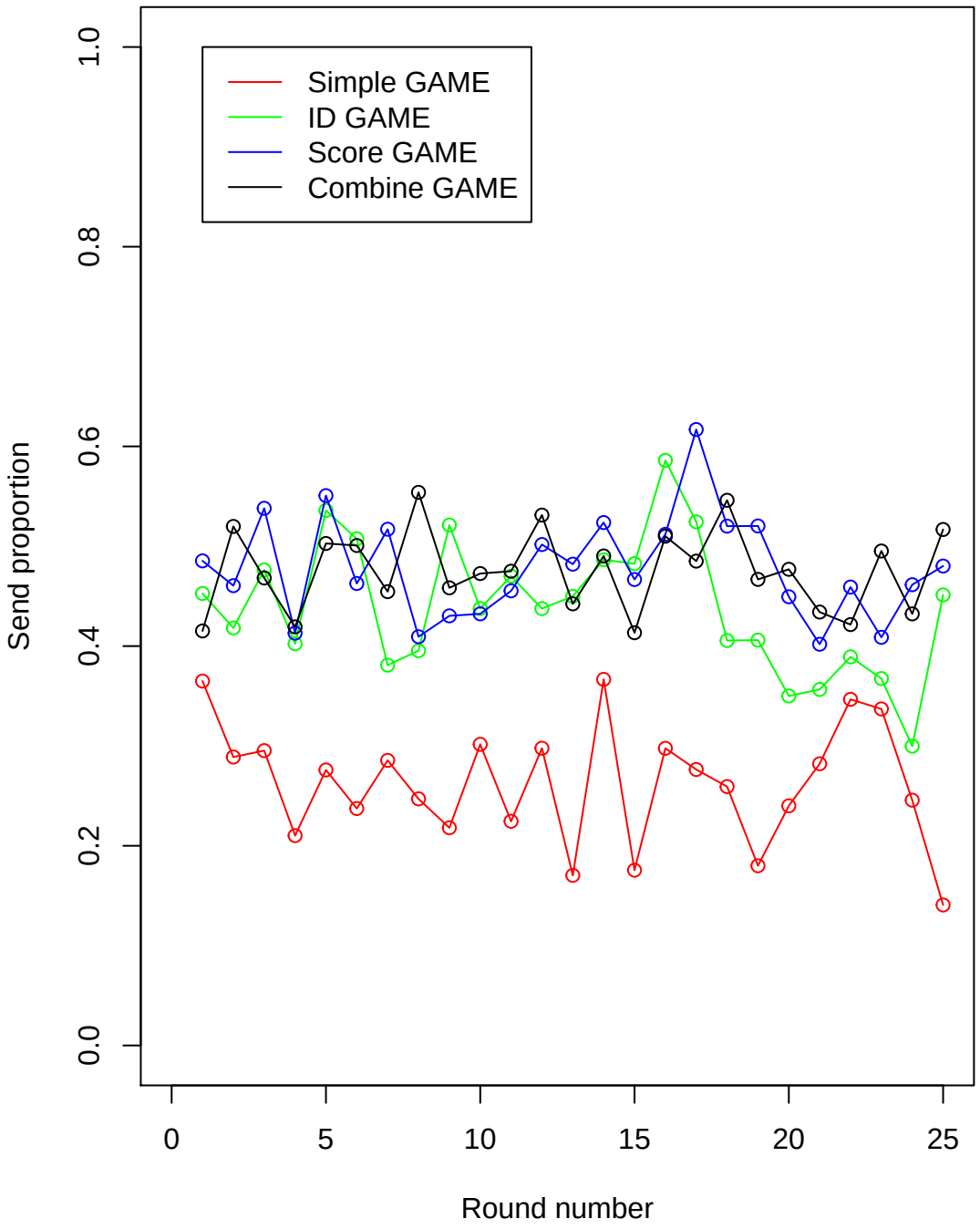


date: Wed 7-Jun-2017

```
## [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
```

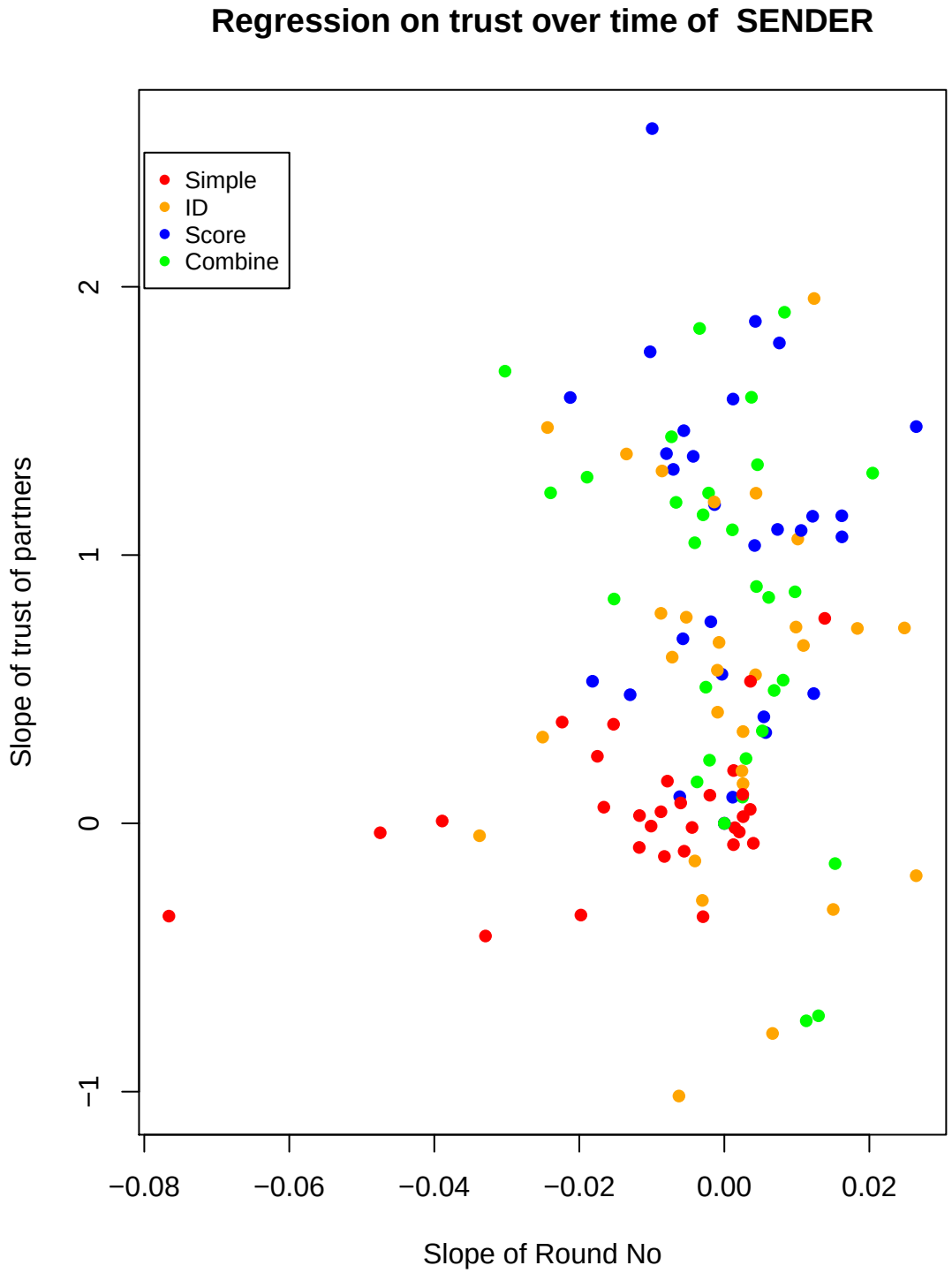
Standard deviation of each user by game

```
## [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

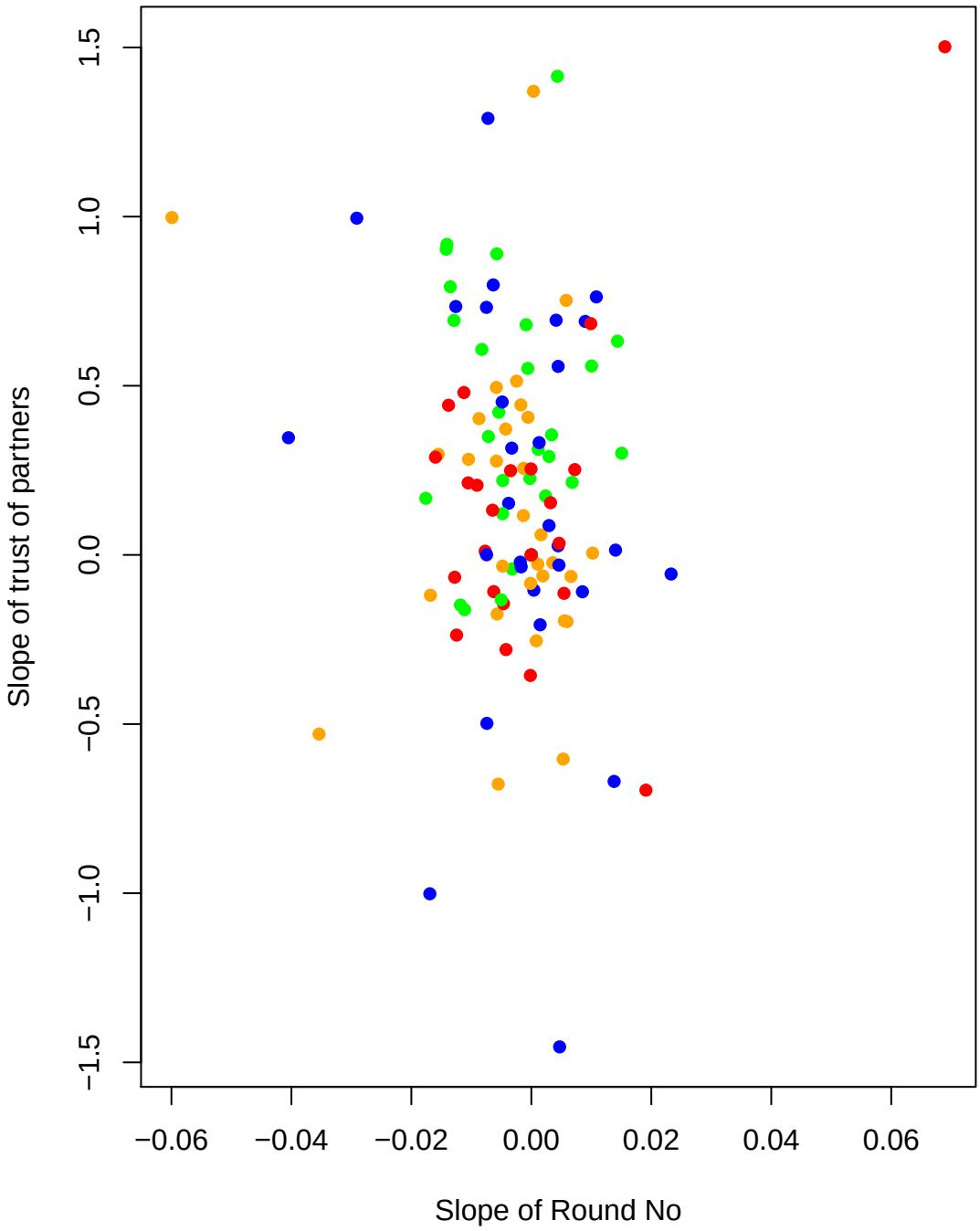
We want to see the adaptation of user on trust score over time.




```
##   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] "---****---"
```




```
##    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] "---****---"
```

Questionaire analysis

In this section, we present the summary of questionaire 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 questionaire 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 questionaire 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 questionarie, 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
```

Consitency 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
```

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

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

```r
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
```

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

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

```r
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
```

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

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

```r
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
```