October 24, 2021

Transmission Game: Supporting Analyses Study 1 (SA1)

Jan K. Woike
Max Planck Institute for Human Development, Berlin, Germany; University of Plymouth, United Kingdom

Sebastian Hafenbrädl
IESE Business School, Barcelona, Spain

Patricia Kanngiesser
Freie Universität Berlin, Germany; University of Plymouth, United Kingdom

Ralph Hertwig
Max Planck Institute for Human Development, Berlin, Germany

Abstract

Transmission Game: Supporting Analyses Study 1 (SA1)

Contents

SA1-1 Data preparation **3**
SA1-1.1 Packages and dataset 3
SA1-1.2 Variables and scale construction 5
SA1-1.3 Transmission game variables 19

SA1-2 Exclusion of participants and sample selection **23**

SA1-3 Relationship between game results and predictors **24**
SA1-3.1 HEXACO-scales 24
SA1-3.2 Other personality scales 25
SA1-3.3 Cognitive variables 26
SA1-3.4 Economic games 27

Jan K. Woike https://orcid.org/0000-0002-6816-121X, Sebastian Hafenbrädl https://orcid.org/0000-0002-5148-766X, Patricia Kanngiesser https://orcid.org/0000-0003-1068-3725, Ralph Hertwig https://orcid.org/0000-0002-9908-9556
Correspondence e-mail: woike@mpib-berlin.mpg.de

SA1-3.5 Single-item personality measures . . . 29
SA1-3.6 Political measures . . . 30
SA1-3.7 Age and health . . . 33
SA1-3.8 Categorical variables . . . 34
SA1-3.8.1 Demographics . . . 34
SA1-3.8.2 Religion and politics . . . 38

SA1-4 Exploratory analysis of relationships between variables 40
SA1-4.1 COVID-19 related scales and game scores . . . 40
SA1-4.2 COVID-19 related scales and HEXACO scales . . . 41
SA1-4.3 COVID-19 related scales and other personality scales . . . 42
SA1-4.4 COVID-19 related scales and cognitive variables . . . 46
SA1-4.5 COVID-19 related scales and economic games . . . 48
SA1-4.6 COVID-19 related scales and single-item personality measures . . . 50
SA1-4.7 COVID-19 related scales and political measures . . . 52
SA1-4.8 COVID-19 related scales and age and health . . . 54
SA1-4.9 Multiple regression models: Predicting game scores . . . 56
SA1-4.9.1 Motivation . . . 57
SA1-4.9.2 Expectation Purple . . . 59
SA1-4.9.3 Percent H-choices . . . 61

List of Figures

SA1-1 Transmission Game results distribution . . . 22
SA1-2 Transmission Game roundwise results . . . 23
SA1-3 HEXACO scales . . . 25
SA1-4 Other personality scales . . . 26
SA1-5 Cognitive variables . . . 27
SA1-6 Economic Games . . . 28
SA1-7 Single-item personality variables . . . 30
SA1-8 Political measures . . . 32
SA1-9 Age and health . . . 34
SA1-10 Gender . . . 35
SA1-11 Education . . . 36
SA1-12 Income . . . 38
SA1-13 Religion . . . 39
SA1-14 Political party . . . 40
SA1-15 COVID-19 scales . . . 41
SA1-16 HEXACO scales . . . 43
SA1-17 Other personality scales . . . 45
SA1-18 Cognitive variables . . . 47
SA1-19 Economic Games . . . 49
SA1-20 Single-item personality variables . . . 51
SA1-21 Political measures . . . 53
SA1-22 Age and health . . . 55

SA1-23 Postquestionnaire motivation 58
SA1-24 Postquestionnaire expectation percent purple 60
SA1-25 Postquestionnaire expectation percent H-choices 62

List of Tables

SA1-1 Error count frequencies in comprenhension check 24

SA1-1 Data preparation

SA1-1.1 Packages and dataset

This document was prepared in Overleaf, as an Rtex file implementing knitr. Any output is generated by R during compilation, and can thus be replicated by entering the same commands referencing the same dataset. Overleaf's R version and selection and versions of packages are not under the user's control. This section demonstrates the R version and the list of packages used for calculations and output generation. https://cran.r-project.org/web/packages/psych/psych.pdf

```
# Loading required libraries
library(foreign)
library(ggplot2)
library(dplyr)

##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyr)
library("purrr")
library("tidyverse")

## Warning in system("timedatectl", intern = TRUE): running command
'timedatectl' had status 1
## - Attaching packages ------------------- tidyverse 1.3.1 -
## v tibble  3.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
## - Conflicts ---------------------- tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library("psych", verbose=TRUE)
```

```
##
## Attaching package:  'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

library("rmarkdown",verbose=TRUE)


library("viridis")

## Loading required package:  viridisLite

library(viridisLite)

# R version
R.version

##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          3
## minor          6.3
## year           2020
## month          02
## day            29
## svn rev        77875
## language       R
## version.string R version 3.6.3 (2020-02-29)
## nickname       Holding the Windsock

# Loading data
fn='TRANSMISSION_GAME_STUDY1_DEIDENTIFIED.sav'
dataS=read.spss(file=fn)
df=data.frame(dataS)

palette2=colorRampPalette(c("#ff7f50", "white", "#2171B5"))

sessionInfo()

## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
```

```
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
##  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
##  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
## [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] viridis_0.6.1     viridisLite_0.4.0 rmarkdown_2.9     psych_2.1.6
##  [5] forcats_0.5.1     stringr_1.4.0     readr_2.0.0       tibble_3.1.3
##  [9] tidyverse_1.3.1   purrr_0.3.4       tidyr_1.1.3       dplyr_1.0.7
## [13] ggplot2_3.3.5     foreign_0.8-76    knitr_1.33
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7         lubridate_1.7.10  lattice_0.20-44   assertthat_0.2.1
##  [5] digest_0.6.27      utf8_1.2.2        R6_2.5.0          cellranger_1.1.0
##  [9] backports_1.2.1    reprex_2.0.0      evaluate_0.14     httr_1.4.2
## [13] pillar_1.6.2       rlang_0.4.11      readxl_1.3.1      rstudioapi_0.13
## [17] munsell_0.5.0      broom_0.7.9       compiler_3.6.3    modelr_0.1.8
## [21] xfun_0.24          pkgconfig_2.0.3   mnormt_2.0.2      tmvnsim_1.0-2
## [25] htmltools_0.5.1.1 tidyselect_1.1.1  gridExtra_2.3     fansi_0.5.0
## [29] crayon_1.4.1       tzdb_0.1.2        dbplyr_2.1.1      withr_2.4.2
## [33] grid_3.6.3         nlme_3.1-152      jsonlite_1.7.2    gtable_0.3.0
## [37] lifecycle_1.0.0    DBI_1.1.1         magrittr_2.0.1    scales_1.1.1
## [41] cli_3.0.1          stringi_1.7.3     fs_1.5.0          xml2_1.3.2
## [45] ellipsis_0.3.2     generics_0.1.0    vctrs_0.3.8       tools_3.6.3
## [49] glue_1.4.2         hms_1.1.0         parallel_3.6.3    colorspace_2.0-2
## [53] rvest_1.0.1        haven_2.4.1
```

SA1-1.2 Variables and scale construction

```
    VarsPost <- c("PostTG_1", "PostTG_2", "PostTG_3",
    "PostTG_4", "PostTG_5")

    FramePost <- df[VarsPost]
```

```
   FramePost <- FramePost %>%
   rename(
    Post1=PostTG_1,Post2=PostTG_2,Post3=PostTG_3,Post4=PostTG_4,
    Post5=PostTG_5 )
   FramePost[] <-data.matrix(FramePost)
```

```
polVars <- c("polCandScale_2", "polCandScale_4")
scoresCandidates <- df[polVars]

scoresCandidates <- scoresCandidates %>%
  rename(
     Trump=polCandScale_2,
     Biden=polCandScale_4
    )
```

«» df$polPosition = fct_relevel(df$polPosition, "slightly-", after = 4)$ @

```
dictVars <- c("DictatorGame", "DictatorExpect")
dictFrame <- df[dictVars]

dictFrame <- dictFrame %>%
  rename(
     Behavior=DictatorGame,
     Expectation=DictatorExpect
    )
```

```
burnVars <- c("MonBurnGame", "MonBurnExpect")
burnFrame <- df[burnVars]

burnFrame <- burnFrame %>%
  rename(
     Behavior=MonBurnGame,
     Expectation=MonBurnExpect
    )
```

```
cctgVars <- c("CCtgi01", "CCtgi02", "CCtgi03",
                  "CCtgi04","CCtgi05","CCtgi06","CCtgi07",
                  "CCtgi08", "CCtgi09","CCtgi10","CCtgi11",
                  "CCtgi12","CCtgi13", "CCtgi14","CCtgi15",
                  "CCtgi16.0" )
```

```
comprehensionTGframe <- df[cctgVars]
scaleComprehensionTG=scoreItems(keys=c(1,1,1,1,1,1,
                            1,1,1,1,1,1,1,1,1,1),
                    items=comprehensionTGframe,totals=TRUE)
```

```
## Warning in scoreItems(keys = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, :  Item= CCtgi02 had no variance and was deleted from the data and the
keys.
## Warning in scoreItems(keys = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, :  Item= CCtgi07 had no variance and was deleted from the data and the
keys.
```

```
ccDiceVars <- c("CCerrorsDice01", "CCerrorsDice02", "CCerrorsDice03",
             "CCerrorsDice04")


comprehensionDiceframe <- df[ccDiceVars]


 scaleComprehensionDice=scoreItems(keys=c(1,1,1,1),
                items=comprehensionDiceframe,totals=TRUE)
```

```
## Warning in scoreItems(keys = c(1, 1, 1, 1), items =
comprehensionDiceframe, :  Item= CCerrorsDice03 had no variance and was
deleted from the data and the keys.
```

```
 comprehensionScales=data.frame(
           scaleComprehensionTG$scores,
           scaleComprehensionDice$scores
            )
comprehensionScales <- comprehensionScales %>%
     rename(
           TG=Scale1,
           Dice=Scale1.1
           )
```

```
 scaleVars <- c("HexBHI_O01", "HexBHI_C02", "HexBHI_A03R",
 "HexBHI_X04R","HexBHI_E05","HexBHI_H06","HexBHI_O07R",
 "HexBHI_C08R", "HexBHI_A09R","HexBHI_X10","HexBHI_E11R",
 "HexBHI_H12R","HexBHI_O13", "HexBHI_C14","HexBHI_A15",
 "HexBHI_X16","HexBHI_E17R","HexBHI_H18R","HexBHI_O19",
 "HexBHI_C20R","HexBHI_A21","HexBHI_X22R","HexBHI_E23",
 "HexBHI_H24R", "HexacoH_06", "HexacoH_12R", "HexacoH_18",
 "HexacoH_24R", "HexacoH_30R", "HexacoH_36", "HexacoH_42R",
 "HexacoH_48R", "HexacoH_54", "HexacoH_60R" )
```

```
  scaleFrame <- df[scaleVars]

  head(scaleFrame$HexBHI_X04R)
```

```
## [1] 4-\nagree             2-\ndisagree          2-\ndisagree
## [4] 1-\nstrongly disagree 2-\ndisagree          1-\nstrongly disagree
## 5 Levels: 1-\nstrongly disagree ... 5-\nstrongly agree
```

```
   scaleFrame <- scaleFrame %>%
    rename(
    bO1p=HexBHI_O01, bC1p=HexBHI_C02, bA1n=HexBHI_A03R,
    bX1n=HexBHI_X04R,bE1p=HexBHI_E05, bH1p=HexBHI_H06,
    bO2n=HexBHI_O07R,bC2n=HexBHI_C08R,bA2n=HexBHI_A09R,
    bX2p=HexBHI_X10,bE2n=HexBHI_E11R, bH2n=HexBHI_H12R,
    bO3p=HexBHI_O13, bC3p=HexBHI_C14, bA3p=HexBHI_A15,
    bX3p=HexBHI_X16,bE3n=HexBHI_E17R, bH3n=HexBHI_H18R,
    bO4p=HexBHI_O19, bC4n=HexBHI_C20R,bA4p=HexBHI_A21,
    bX4n=HexBHI_X22R,bE4p=HexBHI_E23, bH4n=HexBHI_H24R,
    H60_01p=HexacoH_06,H60_02n=HexacoH_12R,H60_03p=HexacoH_18,
    H60_04n=HexacoH_24R,H60_05n=HexacoH_30R,H60_06p=HexacoH_36,
    H60_07n=HexacoH_42R,H60_08n=HexacoH_48R,H60_09p=HexacoH_54,
    H60_10n=HexacoH_60R
  )

 scaleFrame[] <-data.matrix(scaleFrame)

 weightsBHI <-list(scaleBH=c("bH1p","-bH2n","-bH3n","-bH4n"),
  scaleBE = c("bE1p","-bE2n","-bE3n","bE4p"),
  scaleBX = c("-bX1n","bX2p","bX3p","-bX4n"),
  scaleBA =  c("-bA1n","-bA2n","bA3p","bA4p"),
  scaleBC =  c("bC1p","-bC2n","bC3p","-bC4n"),
  scaleBO = c("bO1p","-bO2n","bO3p","bO4p"),
  scaleH60= c("H60_01p","-H60_02n","H60_03p","-H60_04n",
              "-H60_05n","H60_06p","-H60_07n","-H60_08n",
              "H60_09p","-H60_10n")
 )

scoresHEXACO=scoreItems(keys= weightsBHI , items=scaleFrame)
```

```
  scaleReactanceVars <- c("PsyReact_01", "PsyReact_02", "PsyReact_03",
  "PsyReact_04","PsyReact_05","PsyReact_06","PsyReact_07",
  "PsyReact_08", "PsyReact_09", "PsyReact_10", "PsyReact_11")

  scaleReactanceFrame <- df[scaleReactanceVars]
```

```
  scaleReactanceFrame <- scaleReactanceFrame %>%
    rename(
    PR01=PsyReact_01,
    PR02=PsyReact_02,
    PR03=PsyReact_03,
    PR04=PsyReact_04,
    PR05=PsyReact_05,
    PR06=PsyReact_06,
    PR07=PsyReact_07,
    PR08=PsyReact_08,
    PR09=PsyReact_09,
    PR10=PsyReact_10,
    PR11=PsyReact_11
  )


  scaleReactanceFrame[] <-data.matrix(scaleReactanceFrame)
 scoresPR=scoreItems(items=scaleReactanceFrame,
 keys=c(1,1,1,1,1,1,1,1,1,1,1))


scoresPR<-data.frame(scoresPR$scores) %>%
  rename(
    scalePR=Scale1
  )
```

```
  scaleVars <- c("UTIL01_IB01", "UTIL03_IB02", "UTIL01_IB03",
  "UTIL07_IB04","UTIL09_IB05","UTIL02_IH01","UTIL01_IH02",
  "UTIL06_IH03", "UTIL08_IH04")

  scaleFrame <- df[scaleVars]

  scaleFrame <- scaleFrame %>%
    rename(
    IB1p=UTIL01_IB01,
    IB2p=UTIL03_IB02,
    IB3p=UTIL01_IB03,
    IB4p=UTIL07_IB04,
    IB5p=UTIL09_IB05,
    IH1p=UTIL02_IH01,
    IH2p=UTIL01_IH02,
    IH3p=UTIL06_IH03,
    IH4p=UTIL08_IH04
```

```
 )

 scaleFrame[] <-data.matrix(scaleFrame)
  scaleFrameIB=scaleFrame[1:5]
 scaleFrameIH=scaleFrame[6:9]
 scaleKeyIB=c(1,1,1,1,1)
 scaleIB=scoreItems(keys=scaleKeyIB,
 items=scaleFrameIB,totals=FALSE,missing=FALSE)

 scaleKeyIH=c(1,1,1,1)
 scaleIH=scoreItems(keys=scaleKeyIH,
 items =scaleFrameIH,totals=FALSE,missing=FALSE)

scoresOUS<-data.frame(scaleIB$scores, scaleIH$scores) %>%
 rename(
   scaleOUIB=Scale1,
   scaleOUIH=Scale1.1
 )
```

```
   scaleVarsCRT <- c("CRT01", "CRT02", "CRT03",
        "CRT04")
   scaleFrameCRT <- df[scaleVarsCRT]

   scaleFrameCRT$CRT01num <-  as.numeric(scaleFrameCRT$CRT01)
   scaleFrameCRT$CRT02num <-  as.numeric(scaleFrameCRT$CRT02)
   scaleFrameCRT$CRT03num <-  as.numeric(scaleFrameCRT$CRT03)
   scaleFrameCRT$CRT04num <-  as.numeric(scaleFrameCRT$CRT04)

    scaleFrameCRT = subset(scaleFrameCRT,
       select=-c(CRT01,CRT02,CRT03,CRT04))

   scaleFrameCRT$CRT01cat <- recode(scaleFrameCRT$CRT01num,
          "7"="intuitive","8"="correct", .default="other")
   scaleFrameCRT$CRT02cat <- recode(scaleFrameCRT$CRT02num,
         "5"="intuitive","6"="correct", .default="other")
   scaleFrameCRT$CRT03cat <- recode(scaleFrameCRT$CRT03num,
        "4"="intuitive","7"="correct", .default="other")
   scaleFrameCRT$CRT04cat <- recode(scaleFrameCRT$CRT04num,
       "4"="intuitive","5"="correct", .default="other")

   scaleFrameCRT = subset(scaleFrameCRT,
      select=-c(CRT01num,CRT02num,CRT03num,CRT04num))

   scaleFrameCRT$CRT01int <- recode(scaleFrameCRT$CRT01cat,
```

```
    "intuitive"=1, .default=0)
scaleFrameCRT$CRT02int <- recode(scaleFrameCRT$CRT02cat,
    "intuitive"=1, .default=0)
scaleFrameCRT$CRT03int <- recode(scaleFrameCRT$CRT03cat,
    "intuitive"=1, .default=0)
scaleFrameCRT$CRT04int <- recode(scaleFrameCRT$CRT04cat,
    "intuitive"=1, .default=0)

scaleFrameCRT$CRT01correct <- recode(scaleFrameCRT$CRT01cat,
    "correct"=1, .default=0)
scaleFrameCRT$CRT02correct <- recode(scaleFrameCRT$CRT02cat,
    "correct"=1, .default=0)
scaleFrameCRT$CRT03correct <- recode(scaleFrameCRT$CRT03cat,
    "correct"=1, .default=0)
scaleFrameCRT$CRT04correct <- recode(scaleFrameCRT$CRT04cat,
    "correct"=1, .default=0)

scaleFrameCRT = subset(scaleFrameCRT,
     select=-c(CRT01cat,CRT02cat,CRT03cat,CRT04cat))

weightsCRT <-list(CRTscore=c("CRT01correct",
     "CRT02correct","CRT03correct","CRT04correct"),
     CRTintuitive=c("CRT01int",
     "CRT02int","CRT03int","CRT04int"))


scaleCRT=scoreItems(keys=weightsCRT, items =scaleFrameCRT,totals=TRUE)

scoresCRT<-data.frame(scaleCRT$scores)
```

```
scaleVarsEXP <- c("EXP01", "EXP02", "EXP03",
    "EXP04")
scaleFrameEXP <- df[scaleVarsEXP]

scaleFrameEXP$EXP01num <-  as.numeric(scaleFrameEXP$EXP01)
scaleFrameEXP$EXP02num <-  as.numeric(scaleFrameEXP$EXP02)
scaleFrameEXP$EXP03num <-  as.numeric(scaleFrameEXP$EXP03)
scaleFrameEXP$EXP04num <-  as.numeric(scaleFrameEXP$EXP04)

scaleFrameEXP = subset(scaleFrameEXP,
    select=-c(EXP01,EXP02,EXP03,EXP04))

scaleFrameEXP$EXP01corW <- recode(scaleFrameEXP$EXP01num,
    "10"=1,"9"=0.5,"11"=0.5, .default=0)
```

```
scaleFrameEXP$EXP02corW <- recode(scaleFrameEXP$EXP02num,
   "9"=1,"10"=0.5,"8"=0.5, .default=0)
scaleFrameEXP$EXP03corW <- recode(scaleFrameEXP$EXP03num,
    "8"=1,"9"=0.5,"7"=0.5, .default=0)
scaleFrameEXP$EXP04corW <- recode(scaleFrameEXP$EXP04num,
     "5"=1,"6"=0.5,"4"=0.5,  .default=0)

scaleFrameEXP$EXP01corN <- recode(scaleFrameEXP$EXP01num,
    "10"=1, .default=0)
scaleFrameEXP$EXP02corN  <- recode(scaleFrameEXP$EXP02num,
    "9"=1, .default=0)
scaleFrameEXP$EXP03corN  <- recode(scaleFrameEXP$EXP03num,
    "8"=1, .default=0)
scaleFrameEXP$EXP04corN  <- recode(scaleFrameEXP$EXP04num,
    "5"=1, .default=0)

scaleFrameEXP = subset(scaleFrameEXP,
     select=-c(EXP01num ,EXP02num ,EXP03num ,EXP04num ))

weightsEXP <-list(EXPscoreWide=c("EXP01corW",
     "EXP02corW","EXP03corW","EXP04corW"),
     EXPscoreNarrow=c("EXP01corN",
     "EXP02corN","EXP03corN","EXP04corN"))

scaleEXP=scoreItems(keys=weightsEXP, items =scaleFrameEXP,totals=TRUE)
scoresEXP<-data.frame(scaleEXP$scores)
```

```
scaleVarsBNT <- c("BNT01", "BNT02", "BNT03",
     "BNT04")
scaleFrameBNT <- df[scaleVarsBNT]

scaleFrameBNT$BNT01num <-  as.numeric(scaleFrameBNT$BNT01)
scaleFrameBNT$BNT02num <-  as.numeric(scaleFrameBNT$BNT02)
scaleFrameBNT$BNT03num <-  as.numeric(scaleFrameBNT$BNT03)
scaleFrameBNT$BNT04num <-  as.numeric(scaleFrameBNT$BNT04)

scaleFrameBNT = subset(scaleFrameBNT,
    select=-c(BNT01,BNT02,BNT03,BNT04))

 scaleFrameBNT$BNT01cor <- recode(scaleFrameBNT$BNT01num,
    "6"=1, .default=0)
scaleFrameBNT$BNT02cor  <- recode(scaleFrameBNT$BNT02num,
    "8"=1, .default=0)
scaleFrameBNT$BNT03cor  <- recode(scaleFrameBNT$BNT03num,
```

```
      "8"=1, .default=0)
   scaleFrameBNT$BNT04cor  <- recode(scaleFrameBNT$BNT04num,
      "8"=1, .default=0)

   scaleFrameBNT = subset(scaleFrameBNT,
       select=-c(BNT01num ,BNT02num ,BNT03num ,BNT04num ))

   weightsBNT <-list(BNTscorBNT=c("BNT01cor",
       "BNT02cor","BNT03cor","BNT04cor"))
   scaleBNT=scoreItems(keys=weightsBNT, items =scaleFrameBNT,totals=TRUE)
   scoresBNT<-data.frame(scaleBNT$scores)
```

```
scaleVarsSN <- c("SN01", "SN02", "SN03")
scaleFrameSN <- df[scaleVarsSN]

scaleFrameSN$SN01 <-  as.numeric(scaleFrameSN$SN01)
scaleFrameSN$SN02  <-  as.numeric(scaleFrameSN$SN02)
scaleFrameSN$SN03  <-  as.numeric(scaleFrameSN$SN03)

scaleSN=scoreItems(keys=c(1,1,1), items =scaleFrameSN,totals=FALSE)

scoresSN<-data.frame(scaleSN$scores)
```

```
scaleVarsDC <- c("DarkCore01", "DarkCore02", "DarkCore03", "DarkCore04",
"DarkCore05", "DarkCore06", "DarkCore07", "DarkCore08",
"DarkCore09", "DarkCore10","DarkCore11", "DarkCore12",
"DarkCore13", "DarkCore14", "DarkCore15","DarkCore16")
scaleFrameDC <- df[scaleVarsDC]

 scaleFrameDC <- scaleFrameDC %>%
rename(
DC01n=DarkCore01,DC02p=DarkCore02,DC03n=DarkCore03,DC04p=DarkCore04,
DC05n=DarkCore05,DC06p=DarkCore06,DC07n=DarkCore07,DC08p=DarkCore08,
DC09n=DarkCore09,DC10p=DarkCore10,DC11p=DarkCore11,DC12p=DarkCore12,
DC13n=DarkCore13,DC14n=DarkCore14,DC15p=DarkCore15,DC16n=DarkCore16
 )
scaleFrameDC[] <-data.matrix(scaleFrameDC)

weightsDC <-list(scaleDC=c("-DC01n","DC02p","-DC03n","DC04p",
"-DC05n","DC06p","-DC07n","DC08p","-DC09n","DC10p",
 "DC11p","DC12p","-DC13n","-DC14n","DC15p","-DC16n")
)
scaleDC=scoreItems(keys=weightsDC, items =scaleFrameDC,totals=FALSE)
scoresDC<-data.frame(scaleDC$scores)
```

```
scaleVarsCC <- c("ClimCh1", "ClimCh2", "ClimCh3", "ClimCh4", "ClimCh5",
 "ClimCh6")

scaleFrameCC <- df[scaleVarsCC]

 scaleFrameCC <- scaleFrameCC %>%
rename(
CC1p=ClimCh1,CC2n=ClimCh2,CC3n=ClimCh3,CC4n=ClimCh4,
CC5p=ClimCh5,CC6p=ClimCh6  )

scaleFrameCC[] <-data.matrix(scaleFrameCC)

weightsCC <-list(sclClimSkept=c("CC1p","-CC2n","-CC3n","-CC4n",
"CC5p","CC6p")
)
scaleCC=scoreItems(keys=weightsCC, items =scaleFrameCC,totals=FALSE)
scoresCC<-data.frame(scaleCC$scores)
```

```
  scaleVarsSECS <- c("SECScale_1", "SECScale_2", "SECScale_5",
  "SECScale_6", "SECScale_7", "SECScale_8", "SECScale_9",
  "SECScale_10", "SECScale_11", "SECScale_12","SECScale_13",
  "SECScale_14")

  scaleFrameSECS <- df[scaleVarsSECS]

scaleFrameSECS <- scaleFrameSECS %>%
  rename(
  SC01n=SECScale_1,EC01n=SECScale_2,EC02p=SECScale_5,SC02p=SECScale_6,
  SC03p=SECScale_7,EC03p=SECScale_8,SC04p=SECScale_9,SC05p=SECScale_10,
  EC04p=SECScale_11,EC05p=SECScale_12,SC06p=SECScale_13,SC07p=SECScale_14
)

  scaleFrameSECS[] <-data.matrix(scaleFrameSECS)

weightsSECS <-list(sclConsALL=c("-SC01n","-SC02p","EC02p","SC02p","SC03p",
"EC03p","SC04p", "SC05p","EC04p","EC05p","SC06p","SC07p") ,
 sclConsSoc=c("-SC01n","SC02p","SC03p","SC04p","SC05p","SC06p","SC07p"),
  sclConsEcon=c("-EC01n","EC02p","EC03p","EC04p","EC05p")
)

scaleSECS=scoreItems(keys=weightsSECS, items =scaleFrameSECS,totals=FALSE)

## Number of categories should be increased  in order to count frequencies.

scoresSECS<-data.frame(scaleSECS$scores)
```

```
 scaleVarsSVO <- c("SVO1_1", "SVO2_1", "SVO3_1",
"SVO4_1", "SVO5_1","SVO6_1" )

scaleFrameSVO <- df[scaleVarsSVO]

scaleFrameSVO$SVO1_1 <-  as.numeric(scaleFrameSVO$SVO1_1)
scaleFrameSVO$SVO2_1 <-  as.numeric(scaleFrameSVO$SVO2_1)
scaleFrameSVO$SVO3_1 <-  as.numeric(scaleFrameSVO$SVO3_1)
scaleFrameSVO$SVO4_1 <-  as.numeric(scaleFrameSVO$SVO4_1)
scaleFrameSVO$SVO5_1 <-  as.numeric(scaleFrameSVO$SVO5_1)
scaleFrameSVO$SVO6_1 <-  as.numeric(scaleFrameSVO$SVO6_1)

scaleFrameSVO$SVO1self <- recode(scaleFrameSVO$SVO1_1,
    .default=85)
scaleFrameSVO$SVO1other <- recode(scaleFrameSVO$SVO1_1,
    "1"=85,"2"=76,"3"=68,"4"=59,"5"=50, "6"=41,"7"=33,
    "8"=24,"9"=15)

scaleFrameSVO$SVO2self <- recode(scaleFrameSVO$SVO2_1,
   "1"=85,"2"=87,"3"=89,"4"=91,"5"=93,"6"=94,"7"=96,
   "8"=98,"9"=100)
scaleFrameSVO$SVO2other <- recode(scaleFrameSVO$SVO2_1,
    "1"=15,"2"=19,"3"=24,"4"=28,"5"=33, "6"=37,"7"=41,
    "8"=46,"9"=50)

scaleFrameSVO$SVO3self <- recode(scaleFrameSVO$SVO3_1,
    "1"=50,"2"=54,"3"=59,"4"=63,"5"=68,"6"=72,"7"=76,
    "8"=81,"9"=85)
scaleFrameSVO$SVO3other <- recode(scaleFrameSVO$SVO3_1,
    "1"=100,"2"=98,"3"=96,"4"=94,"5"=93, "6"=91,"7"=89,
    "8"=87,"9"=85)

scaleFrameSVO$SVO4self <- recode(scaleFrameSVO$SVO4_1,
    "1"=50,"2"=54,"3"=59,"4"=63,"5"=68,"6"=72,"7"=76,
    "8"=81,"9"=85)
scaleFrameSVO$SVO4other <- recode(scaleFrameSVO$SVO4_1,
    "1"=100,"2"=89,"3"=79,"4"=68,"5"=58, "6"=47,"7"=36,
    "8"=26,"9"=15)

scaleFrameSVO$SVO5self <- recode(scaleFrameSVO$SVO5_1,
   "1"=100,"2"=94,"3"=88,"4"=81,"5"=75,"6"=69,"7"=63,
   "8"=56,"9"=50)
scaleFrameSVO$SVO5other <- recode(scaleFrameSVO$SVO5_1,
    "1"=50,"2"=56,"3"=63,"4"=69,"5"=75, "6"=81,"7"=88,
```

```
        "8"=94,"9"=100)

    scaleFrameSVO$SVO6self <- recode(scaleFrameSVO$SVO6_1,
       "1"=100,"2"=98,"3"=96,"4"=94,"5"=93,"6"=91,"7"=89,
       "8"=87,"9"=85)
    scaleFrameSVO$SVO6other <- recode(scaleFrameSVO$SVO6_1,
        "1"=50,"2"=54,"3"=59,"4"=63,"5"=68, "6"=72,"7"=76,
        "8"=81,"9"=85)

    scaleFrameSVO = subset(scaleFrameSVO,
        select=-c(SVO1_1,SVO2_1,SVO3_1,SVO4_1,SVO5_1,SVO6_1)
        )

    scaleFrameSVO[] <-data.matrix(scaleFrameSVO)

    sapply(scaleFrameSVO, function(x) c(mean=mean(x), var=var(x), sd=sd(x)))
```

```
##      SVO1self SVO1other  SVO2self SVO2other  SVO3self SVO3other  SVO4self
## mean       85  81.68000 99.400000 48.610000 81.990000 86.300000  68.54000
## var         0 197.41172  6.303030 34.502929 89.363535 16.393939 117.05899
## sd          0  14.05033  2.510584  5.873919  9.453229  4.048943  10.81938
##      SVO4other  SVO5self SVO5other  SVO6self SVO6other
## mean  55.24000  83.34000  66.72000 88.860000  75.99000
## var  686.93172 143.82263 143.81980 37.495354 205.64636
## sd    26.20938  11.99261  11.99249  6.123345  14.34038
```

```
    #"SVO1self" has no variance (everyone chose 85)

    weightsSVO <-list(sclSVOself=c("SVO2self","SVO3self",
    "SVO4self","SVO5self","SVO6self"), sclSVOother=c("SVO1other","SVO2other","SVO3other","S
    "SVO5other","SVO6other")
    )

  meanSVOinc=scoreItems(keys=weightsSVO, items =scaleFrameSVO,totals=FALSE)
```

```
## Number of categories should be increased  in order to count frequencies.
```

```
  sumSVOscores=meanSVOinc$scores
  # 85 was deleted due to lack of variance and is artificially added again
  meanSVOself=5/6*sumSVOscores[,1]+1/6*85
  meanSVOother=sumSVOscores[,2]
  angleSVO=atan(   (meanSVOother -50) / (meanSVOself-50) )* 90/ 1.57079632679
  scoresSVO<-data.frame(angleSVO)
```

```
     scaleVarsCVW <- c("COV_WORyoursel", "COV_WORfinancial",
     "COV_WORotherpeople","COV_WOReconomy", "COV_WORDemocracy" )

    scaleFrameCVW <- df[scaleVarsCVW]

  scaleFrameCVW <- scaleFrameCVW %>%
    rename(
    CVW1=COV_WORyoursel,CVW2=COV_WORfinancial,
    CVW3=COV_WORotherpeople,CVW4=COV_WOReconomy,CVW5=COV_WORDemocracy )

    scaleFrameCVW[] <-data.matrix(scaleFrameCVW)

 weightsCVW <-list(sclCOVWorry=c("CVW1","CVW2","CVW3","CVW4","CVW5")
 )

  scaleCVW=scoreItems(keys=weightsCVW, items =scaleFrameCVW,totals=FALSE)

## Number of categories should be increased  in order to count frequencies.

  scoresCVW<-data.frame(scaleCVW$scores)
```

```
     scaleVarsCVCP <- c("COV_guidelines", "COV_mask", "COV_meet" )

    scaleFrameCVCP <- df[scaleVarsCVCP]

  scaleFrameCVCP <- scaleFrameCVCP %>%
    rename(
    CVCP1p=COV_guidelines,CVCP2p=COV_mask,CVCP3n=COV_meet
    )

    scaleFrameCVCP[] <-data.matrix(scaleFrameCVCP)

 weightsCVCP <-list(sclCompliance=c("CVCP1p","CVCP2p","-CVCP3n")
 )

  scaleCVCP=scoreItems(keys=weightsCVCP, items =scaleFrameCVCP,
  totals=FALSE)

## Number of categories should be increased  in order to count frequencies.

 scoresCVCP<-data.frame(scaleCVCP$scores)
```

```
    scaleVarsCVSD <- c("COV_distancing_1", "COV_distancing_2",
    "COV_distancing_3","COV_distancing_4", "COV_distancing_5",
    "COV_distancing_6" )

   scaleFrameCVSD <- df[scaleVarsCVSD]

 scaleFrameCVSD <- scaleFrameCVSD %>%
   rename(
   CVSD1n=COV_distancing_1,CVSD2n=COV_distancing_2,CVSD3p=COV_distancing_3,
   CVSD4n=COV_distancing_4,CVSD5p=COV_distancing_5,CVSD6n=COV_distancing_6)

   scaleFrameCVSD[] <-data.matrix(scaleFrameCVSD)

weightsCVSD <-list(sclCOVSocDist=c("-CVSD1n","-CVSD2n","CVSD3p",
"-CVSD4n","CVSD5p","-CVSD6n")
)

scaleCVSD=scoreItems(keys=weightsCVSD, items =scaleFrameCVSD,totals=FALSE)
scoresCVSD<-data.frame(scaleCVSD$scores)
```

```
    scaleVarsCVMI <- c("COV_statem1_1", "COV_statem1_2", "COV_statem1_3",
   "COV_statem1_4", "COV_statem1_5","COV_statem1_6",
   "COV_statem1_8","COV_statem2_1","COV_statem2_2","COV_statem2_5")

   scaleFrameCVMI <- df[scaleVarsCVMI]

 scaleFrameCVMI  <- scaleFrameCVMI  %>%
   rename(
   CVMI01p = COV_statem1_1, CVMI02p = COV_statem1_2, CVMI03p = COV_statem1_3,
   CVMI04p = COV_statem1_4, CVMI05p = COV_statem1_5, CVMI06p = COV_statem1_6,
   CVMI07n = COV_statem1_8, CVMI08n = COV_statem2_1, CVMI09p = COV_statem2_2,
   CVMI10p = COV_statem2_5 )

   scaleFrameCVMI[] <-data.matrix(scaleFrameCVMI)

weightsCVMI <-list(sclMisinform=c("CVMI01p",
"CVMI02p","CVMI03p","CVMI04p",
"CVMI05p", "CVMI06p","-CVMI07n",
"-CVMI08n", "CVMI09p", "CVMI10p" )
)

 scaleCVMI=scoreItems(keys=weightsCVMI, items =scaleFrameCVMI,totals=FALSE)
 scoresCVMI<-data.frame(scaleCVMI$scores)
```

```
     scaleVarsCVTO <- c("COV_tradeoffs_1", "COV_tradeoffs_2", "COV_tradeoffs_3",
    "COV_tradeoffs_4", "COV_tradeoffs_5", "COV_tradeoffs_6" )

    scaleFrameCVTO <- df[scaleVarsCVTO]

  scaleFrameCVTO <- scaleFrameCVTO %>%
    rename(
    CVTO1=COV_tradeoffs_1,CVTO2=COV_tradeoffs_2,CVTO3=COV_tradeoffs_3,
    CVTO4=COV_tradeoffs_4,CVTO5=COV_tradeoffs_5, CVTO6=COV_tradeoffs_6 )

    scaleFrameCVTO[] <-data.matrix(scaleFrameCVTO)

weightsCVTO <-list(sclTradeoffs=c("CVTO1","CVTO2","CVTO3","CVTO4","CVTO5","CVTO6")
)


 scaleCVTO=scoreItems(keys=weightsCVTO, items =scaleFrameCVTO,totals=FALSE)
 scoresCVTO<-data.frame(scaleCVTO$scores)
```

```
    scaleVarsDICE <- c("prom1Eyes", "prom2Eyes", "prom3Eyes",
  "prom4Eyes", "prom5Eyes", "prom6Eyes", "prom7Eyes",
  "prom8Eyes", "prom9Eyes", "prom10Eyes", "prom11Eyes",
  "prom12Eyes", "prom13Eyes", "prom14Eyes", "prom15Eyes")

 scaleFrameDICE <- df[scaleVarsDICE]

 scaleFrameDICE <- scaleFrameDICE %>%
   rename(
   DIE01=prom1Eyes,DIE02=prom2Eyes,DIE03=prom3Eyes,
   DIE04=prom4Eyes,DIE05=prom5Eyes,DIE06=prom6Eyes,
   DIE07=prom7Eyes,DIE08=prom8Eyes,DIE09=prom9Eyes,
   DIE10=prom10Eyes,DIE11=prom11Eyes,DIE12=prom12Eyes,
   DIE13=prom13Eyes,DIE14=prom14Eyes,DIE15=prom15Eyes
   )

   scaleFrameDICE[] <-data.matrix(scaleFrameDICE)

 scaleDICE=scoreItems(keys=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
 items =scaleFrameDICE,totals=TRUE)

 scoresDICE<-data.frame(scaleDICE$scores-52.5)
```

SA1-1.3 Transmission game variables

```
  scaleVarsTG <- c("TG_R01Choice", "TG_R02Choice", "TG_R03Choice",
   "TG_R04Choice", "TG_R05Choice", "TG_R06Choice", "TG_R07Choice",
   "TG_R08Choice", "TG_R09Choice", "TG_R10Choice", "TG_R11Choice",
   "TG_R12Choice", "TG_R13Choice", "TG_R14Choice", "TG_R15Choice",
   "TG_R16Choice", "TG_R17Choice", "TG_R18Choice", "TG_R19Choice",
   "TG_R20Choice", "TG_R21Choice", "TG_R22Choice", "TG_R23Choice",
   "TG_R24Choice", "TG_R25Choice"
   )

  scaleFrameTG <- df[scaleVarsTG]

  scaleFrameTG <- scaleFrameTG %>%
    rename(
    R01=TG_R01Choice,R02=TG_R02Choice,R03=TG_R03Choice,
    R04=TG_R04Choice,R05=TG_R05Choice,R06=TG_R06Choice,
    R07=TG_R07Choice,R08=TG_R08Choice,R09=TG_R09Choice,
    R10=TG_R10Choice,R11=TG_R11Choice,R12=TG_R12Choice,
    R13=TG_R13Choice,R14=TG_R14Choice,R15=TG_R15Choice,
    R16=TG_R16Choice,R17=TG_R17Choice,R18=TG_R18Choice,
    R19=TG_R19Choice,R20=TG_R20Choice,R21=TG_R21Choice,
    R22=TG_R22Choice,R23=TG_R23Choice,R24=TG_R24Choice,
    R25=TG_R25Choice
    )

   scaleFrameTG[] <-data.matrix(scaleFrameTG)

  scaleTG=scoreItems(keys=c(1,1,1,1,1,1,1,1,1,1,
         1,1,1,1,1,1,1,1,1,1,
         1,1,1,1,1 ),
  items =scaleFrameTG,totals=TRUE)

#categories are scored 1 for 8 points and 2 for 40 points
  scoresTG<-data.frame(8*25+(scaleTG$scores-25)*32)

  summary(scoresTG)

##      Scale1
##  Min.   : 200.0
##  1st Qu.: 264.0
##  Median : 424.0
##  Mean   : 489.6
##  3rd Qu.: 680.0
##  Max.   :1000.0

  head(scoresTG)
```

```
##   Scale1
## 1   1000
## 2    360
## 3    264
## 4    552
## 5    616
## 6    392
```

```
  scale_meanTG =summarise_all(scoresTG,mean)
  TGdf=data.frame(scale_mean=t(summarise_all(scoresTG,mean)),
          key=names(scoresTG))

  scoresTG %>%
    keep(is.numeric) %>%
      gather() %>%
        ggplot(aes(value)) +
         facet_wrap(~ key, ncol=2) +
         geom_histogram(aes(y =..count..), color="#000044",
                fill="white",bins=26)
```

```
roundwiseTG=pivot_longer(
  cols=starts_with("R"),
   data=scaleFrameTG,
   names_to="Round",
  names_prefix="R"
 )



roundwiseTG <- roundwiseTG %>%
      mutate(partic=rep(seq(from=1,to=100),each=25))

roundwiseTG$Round <- as.numeric(roundwiseTG$Round)

head(roundwiseTG)

## # A tibble: 6 x 3
##   Round value partic
##   <dbl> <int>  <int>
## 1     1     2      1
## 2     2     2      1
## 3     3     2      1
## 4     4     2      1
```

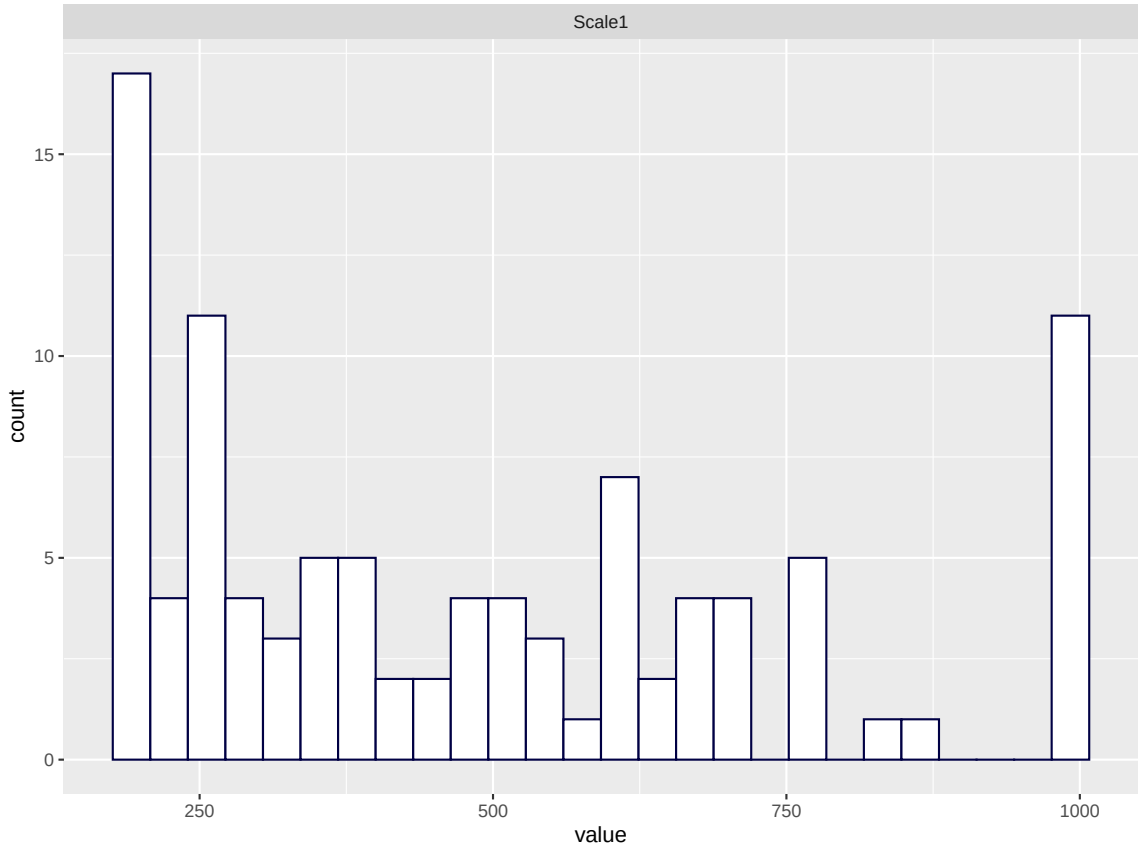

Figure SA1-1
Transmission Game results distribution

```
## 5     5     2     1
## 6     6     2     1

plotdata= roundwiseTG %>%
   group_by(Round) %>%
   summarize(counted=sum(value)-100,
       CIlower=100*prop.test(sum(value)-100, 100)$conf.int[1],
       CIupper=100*prop.test(sum(value)-100, 100)$conf.int[2])
```

```
ggplot(plotdata, aes(x = Round, y = counted)) +
  ylim(0,100) +
  ylab("Percentage of H-choices")+
  geom_line(color="grey75") +
  geom_linerange(aes(ymax = CIupper, ymin = CIlower),color="grey45",size=1.5)+
  geom_point(size = 6,color="blue")
```

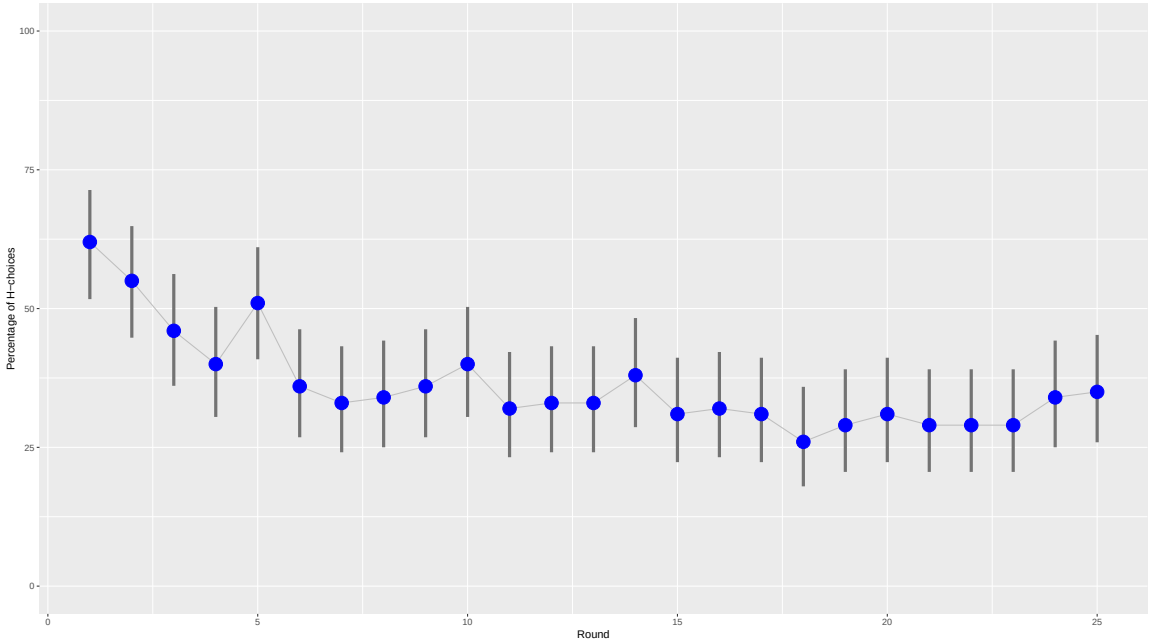

Figure SA1-2
Transmission Game roundwise results: error bars mark the 95-percent CI for percentages

SA1-2 Exclusion of participants and sample selection

A total of 131 participants started the task. Only 106 participants entered the study: Two participants were flagged due to a suspicious IP-address (possible proxy-use or VPN server). Both results were obtained by two independent IP-checkers. Three participants did not respond to the attention check questions, 20 participants failed at least one of the two checks in the beginning and did not enter the study.Of those participants who started, 4 stopped their participation before the game and during the game instructions, 2 did not continue after having completed the game. We analyzed the data for data quality after 100 participants had completed the game, focussing on the number of errors in comprehension checks: Two participants had at least double the number of errors in comprehension checks than the highest number observed in the 98 remaining participants, namely 50 errors and 71 errors, which indicated a very low degree of attention. This degree attention was considered insufficient, even allowing for the fact that the survey demanded correct responses to proceed. These two cases were thus replaced by two additional participants (bringing the initial total to 133), collected after this inspection. The number of errors in the remaining sample is summarized in Table SA1-1. For study 2, a cutoff of 25 errors was hard-coded into the survey so that participants surpassing 25 errors were not able to start the game. This eliminated the researcher's degree of freedom that was admittedly still present in Study 1 (we decided against excluding the borderline case, namely the participant with 25 errors in Study 1) .

Table SA1-1
Error count frequencies in comprenhension check

Errors	frequency
0	39
1	25
2	10
3	10
4	3
5	2
6	1
7	3
8	1
9	2
10	1
14	2
25	1

SA1-3 Relationship between game results and predictors

SA1-3.1 HEXACO-scales

```
corData=data.frame(
  scoresHEXACO$scores,
  scoresTG
)


corData <- corData %>%
  rename(
    Hs=scaleBH,
    Es=scaleBE,
    Xs=scaleBX,
    As=scaleBA,
    Cs=scaleBC,
    Os=scaleBO,
    Hl=scaleH60,
    TGscore=Scale1
  )

pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="blue",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

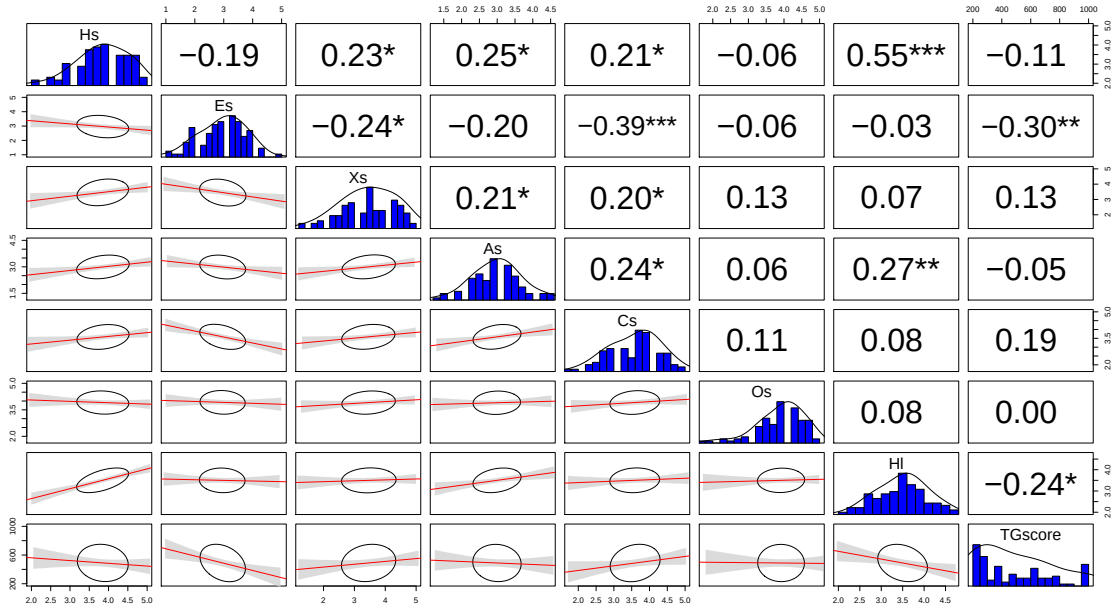

Figure SA1-3
HEXACO scales: Correlations with game score

SA1-3.2 Other personality scales

```
corData=data.frame(
  scoresDC,
  scoresPR,
  scoresOUS,
  scoresTG
)

corData <- corData %>%
  rename(
    DarkCore=scaleDC,
    PsychReactance=scalePR,
    ImpBeneficence=scaleOUIB,
    InstrHarm=scaleOUIH,
    TGscore=Scale1
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#4b7a42",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
```

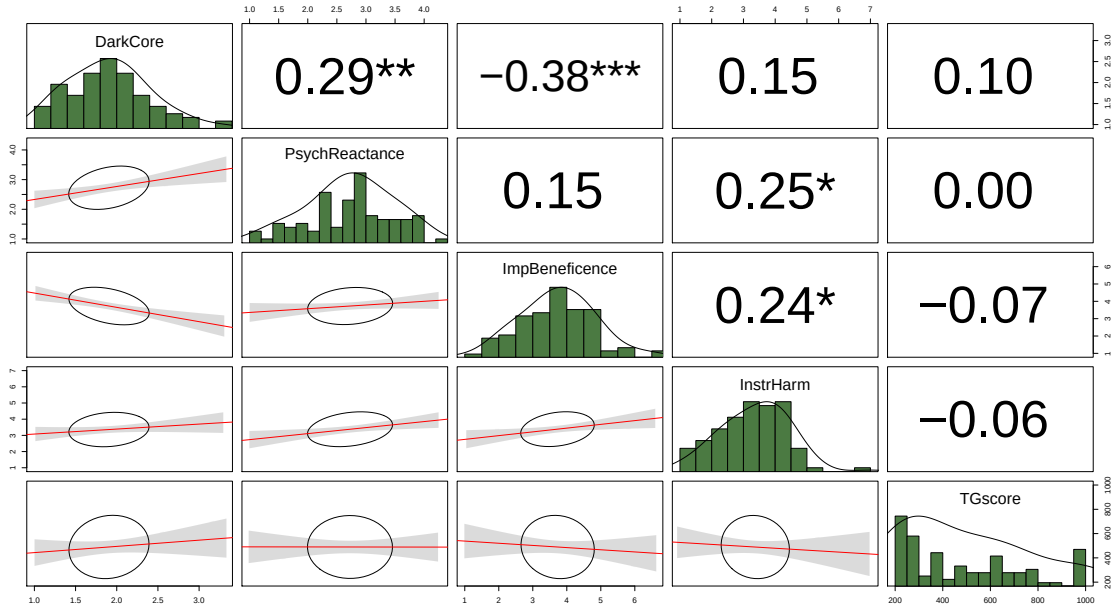

Figure SA1-4
Other personality scales: Correlations with game score

```
              stars=TRUE,ci=TRUE,alpha=.05)
```

SA1-3.3 Cognitive variables

```
corData=data.frame(
  scoresCRT,
  scoresBNT,
  scoresEXP,
  scoresSN,
  scoresTG
)


corData <- corData %>%
  rename(
    CRT=CRTscore,
    CRTi=CRTintuitive,
    BNT=BNTscorBNT,
    EXPw=EXPscoreWide,
    EXPn=EXPscoreNarrow,
    SN=Scale1,
    TGscore=Scale1.1
```

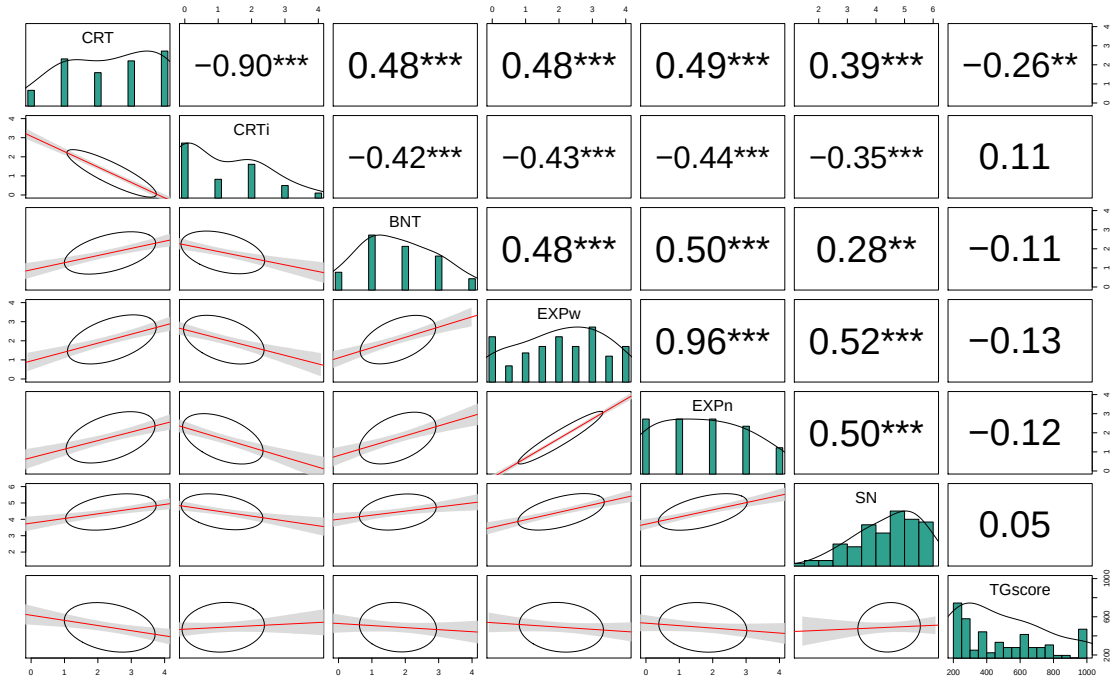

Figure SA1-5
Cognitive variables: Correlations with game score

```
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#30a18e",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

SA1-3.4 Economic games

```
corData=data.frame(
  df$DictatorGame,
  df$DictatorExpect,
  df$MonBurnGame,
  df$MonBurnExpect,
  scoresDICE,
  angleSVO,
  scoresTG
)
```

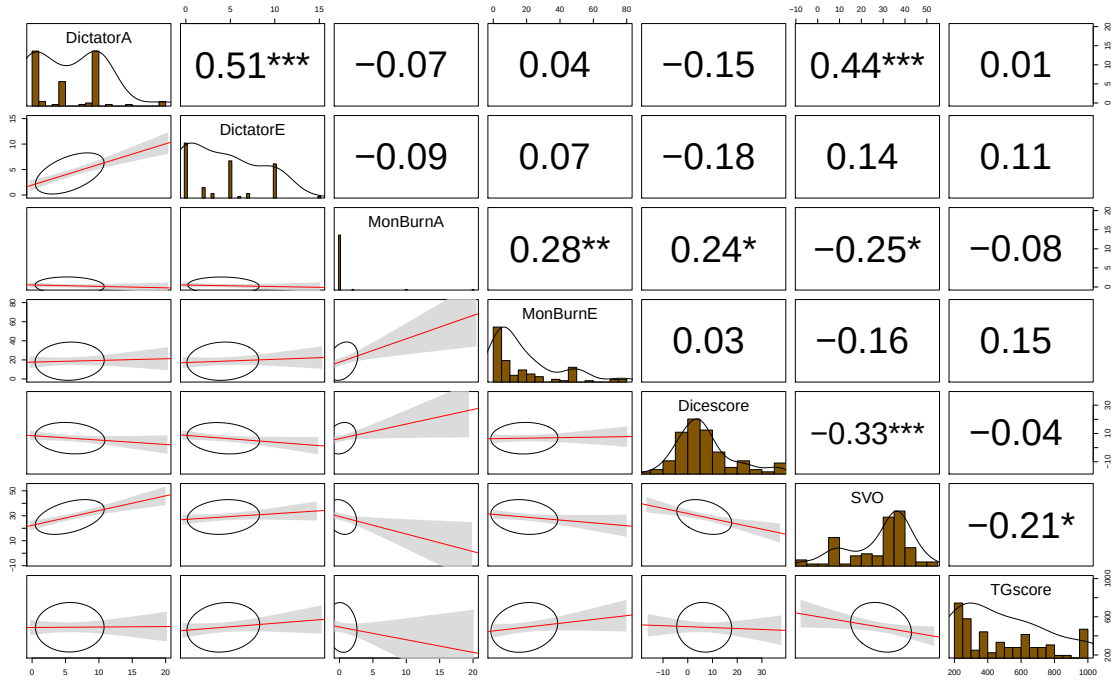

Figure SA1-6
Economic Games: Correlations with game score

```
corData <- corData %>%
  rename(
    DictatorA=df.DictatorGame,
    DictatorE=df.DictatorExpect,
    MonBurnA=df.MonBurnGame,
    MonBurnE=df.MonBurnExpect,
    Dicescore=Scale1,
    SVO=angleSVO,
    TGscore=Scale1.1
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#825503",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

SA1-3.5 Single-item personality measures

```
# variables are adjusted to R's conversion defaults

corData=data.frame(
  as.numeric(df$RTGeneral)-1,
  as.numeric(df$TRUST)-1,
  as.numeric(df$TIMEPREFERENCE)-1,
  as.numeric(df$IMPATIENCE)-1,
  as.numeric(df$IMPULSIVENESS)-1,
  scoresTG
)



# variable names are adjusted to R's variable naming defaults....

corData <- corData %>%
  rename(
    RiskPreference=as.numeric.df.RTGeneral....1,
    Trust=as.numeric.df.TRUST....1,
    TimePreference=as.numeric.df.TIMEPREFERENCE....1,
    Impatience=as.numeric.df.IMPATIENCE....1,
    Impulsiveness=as.numeric.df.IMPULSIVENESS....1,
    TGscore=Scale1
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#1f0b52",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

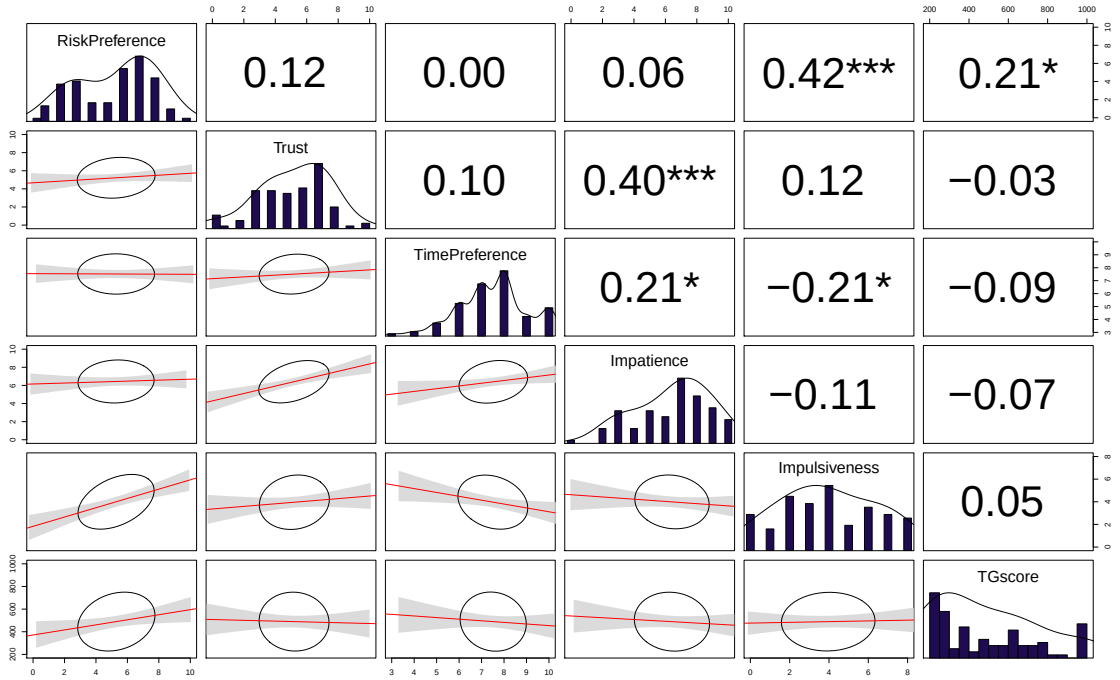

Figure SA1-7
Single-item personality variables: Correlations with game score

SA1-3.6 Political measures

```
corData=data.frame(
 as.numeric(df$polPosition),
 scoresCandidates,
 scoresCC,
 scoresSECS,
 scoresTG
)


corData <- corData %>%
  rename(
    Conservatism=as.numeric.df.polPosition.,
    PrefTrump=Trump,
    PrefBiden=Biden,
    ClimateSkepticism=sclClimSkept,
    SECS=sclConsALL,
    SECSsocial=sclConsSoc,
    SECSeconomic=sclConsEcon,
    TGscore=Scale1
```

```
  )

pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#7f07b3",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

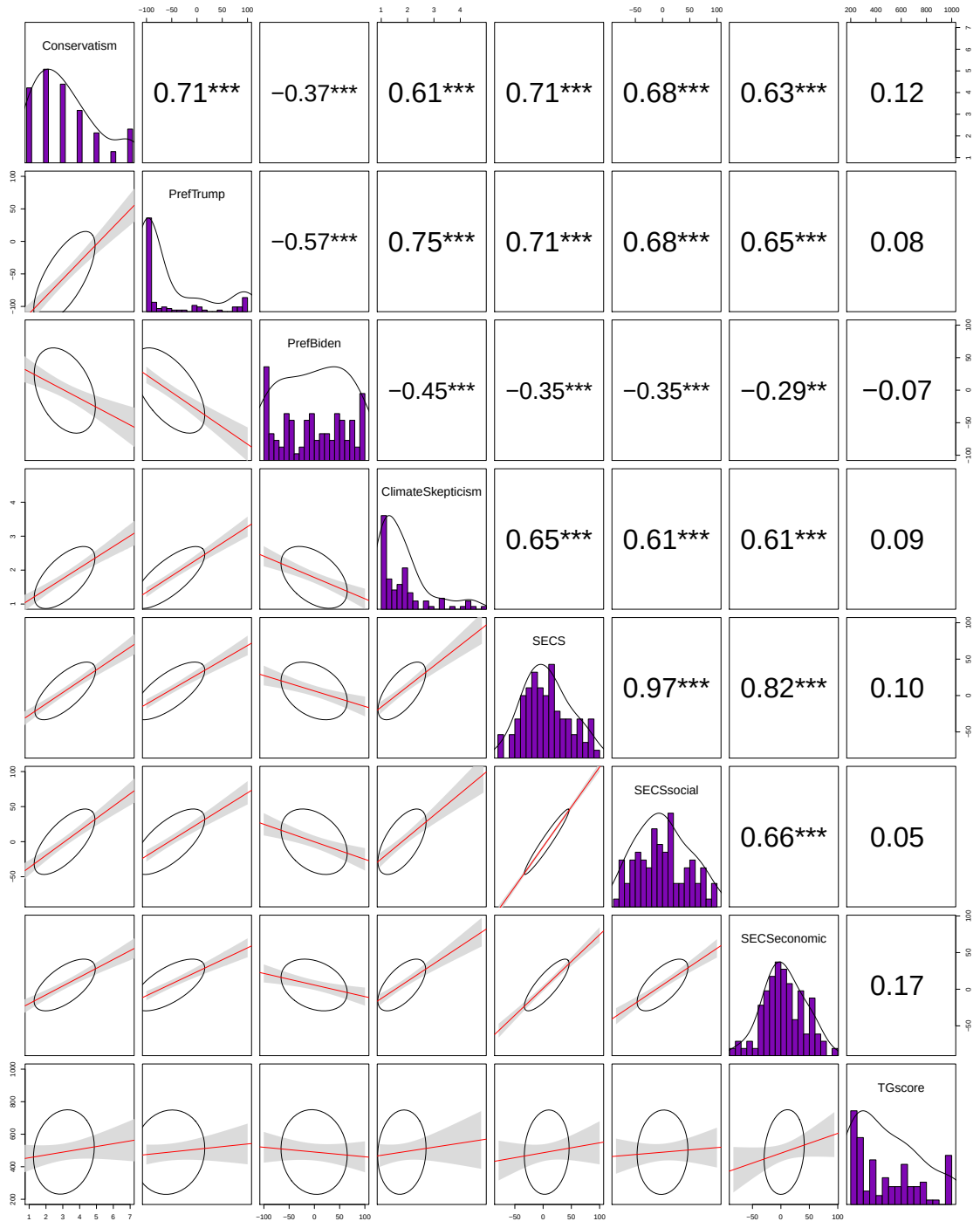

Figure SA1-8

Political measures: Correlations with game score

SA1-3.7 Age and health

```
corData=data.frame(
  df$demo01Age,
  df$GenHealth,
  df$SLExp_1,
  SubjYearsLeft=df$SLExp_1-df$demo01Age,
  scoresTG
)



corData <- corData %>%
  rename(
    Age=df.demo01Age,
    GeneralHealth=df.GenHealth,
    SubjLifeExpect=df.SLExp_1,
    TGscore=Scale1
  )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#990612",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

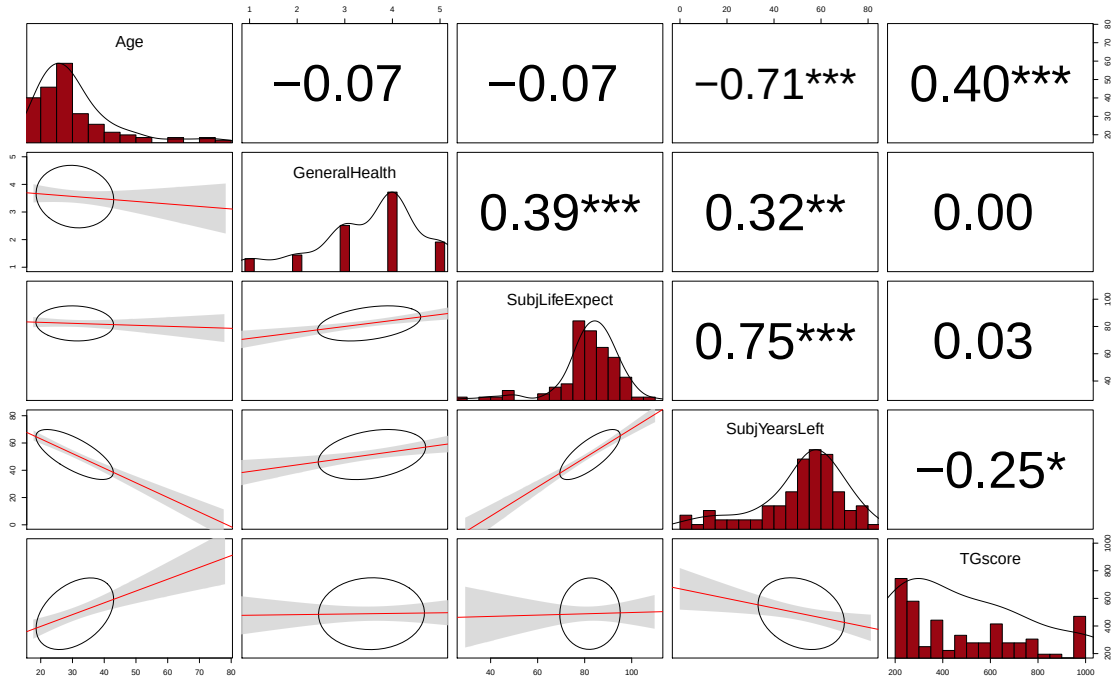

Figure SA1-9
Age and health: Correlations with game score

SA1-3.8 Categorical variables

SA1-3.8.1 Demographics

```
anovaFrame=data.frame(
  gender=df$demo01Gender,
  scoresTG
)

res.aov <- aov(Scale1 ~ gender, data=anovaFrame)
summary(res.aov)

library(dplyr)

group_by(anovaFrame, gender) %>%
  summarise(
    count = n(),
    mean = mean(Scale1, na.rm = TRUE),
    sd = sd(Scale1, na.rm = TRUE)
  )
```

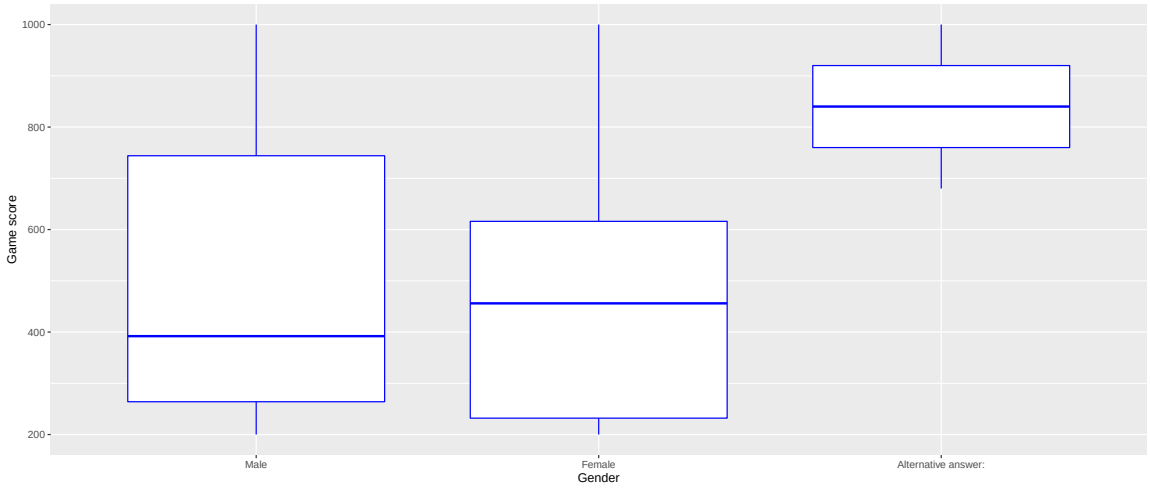

Figure SA1-10
Gender: Correlations with game score

```
ggplot(anovaFrame, aes(x = gender, y = Scale1 )) +
  geom_boxplot(color="blue")+labs(x="Gender",y="Game score")
```

```
##             Df  Sum Sq Mean Sq F value Pr(>F)
## gender       2  338575  169287   2.583 0.0807 .
## Residuals   97 6357105   65537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## # A tibble: 3 x 4
##   gender              count  mean    sd
##   <fct>               <int> <dbl> <dbl>
## 1 Male                   55  509.  276.
## 2 Female                 43  449.  228.
## 3 Alternative answer:     2  840   226.
```

Gender.

```
anovaFrame=data.frame(
  education=df$demo02Education,
  scoresTG
)

res.aov <- aov(Scale1 ~ education, data=anovaFrame)
summary(res.aov)
```

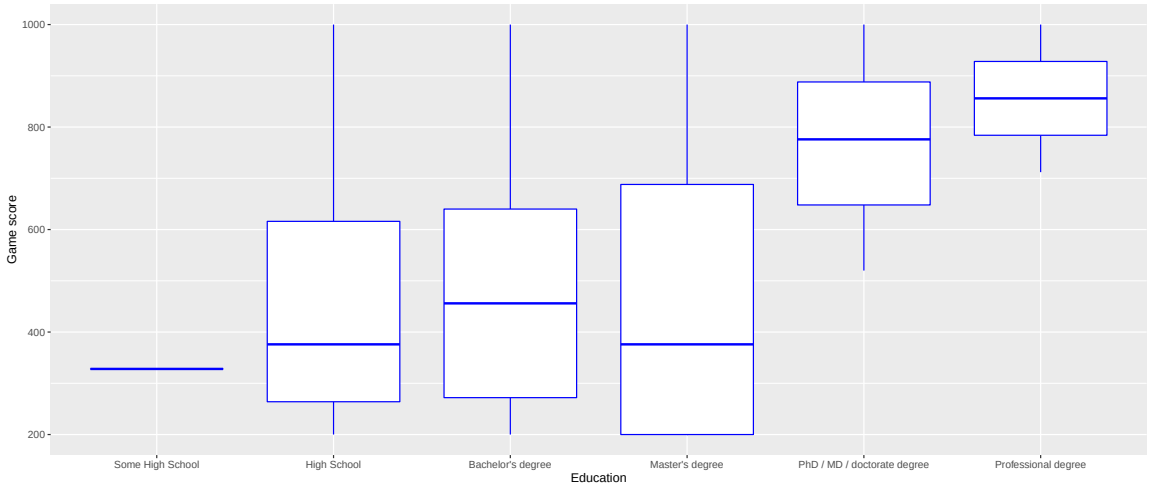

Figure SA1-11
Education: Correlations with game score

```
library(dplyr)

group_by(anovaFrame, education) %>%
  summarise(
    count = n(),
    mean = mean(Scale1, na.rm = TRUE),
    sd = sd(Scale1, na.rm = TRUE)
  )


ggplot(anovaFrame, aes(x = education, y = Scale1 )) +
  geom_boxplot(color="blue")+labs(x="Education",y="Game score")
```

```
##             Df  Sum Sq Mean Sq F value Pr(>F)
## education    5  577350  115470   1.774  0.126
## Residuals   94 6118330   65089
## # A tibble: 6 x 4
##   education                   count  mean    sd
##   <fct>                       <int> <dbl> <dbl>
## 1 Some High School                1  328    NA
## 2 High School                    44  464   255.
## 3 Bachelor's degree              34  499.  255.
## 4 Master's degree                16  452   261.
## 5 PhD / MD / doctorate degree     3  765.  240.
## 6 Professional degree             2  856   204.
```

Education.

```
anovaFrame=data.frame(
  income=df$demo03Income,
  scoresTG
)

res.aov <- aov(Scale1 ~ income, data=anovaFrame)
summary(res.aov)

library(dplyr)

group_by(anovaFrame, income) %>%
  summarise(
    count = n(),
    mean = mean(Scale1, na.rm = TRUE),
    sd = sd(Scale1, na.rm = TRUE)
  )


ggplot(anovaFrame, aes(x = income, y = Scale1 )) +
  geom_boxplot(color="blue")+labs(x="Income",y="Game score")
```

```
##             Df  Sum Sq Mean Sq F value Pr(>F)
## income       4  556805  139201   2.154 0.0801 .
## Residuals   95 6138875   64620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## # A tibble: 5 x 4
##   income             count  mean    sd
##   <fct>              <int> <dbl> <dbl>
## 1 Less than $25,000     17  435.  262.
## 2 $25,000-$50,000       24  581.  278.
## 3 >$50,000-$100,000     32  410   210.
## 4 >$100,000-$200,000    21  515.  290.
## 5 More than $200,000     6  611.  205.
```

Household income (2019).

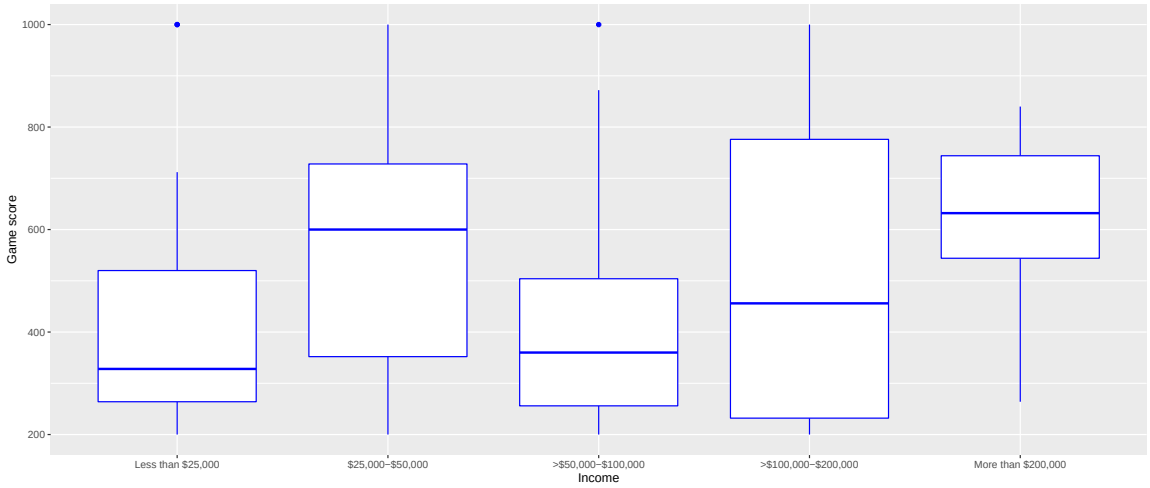

Figure SA1-12
Income: Correlations with game score

SA1-3.8.2 Religion and politics

```
anovaFrame=data.frame(
  df$religion1,
  scoresTG
)

res.aov <- aov(Scale1 ~ df.religion1, data=anovaFrame)
summary(res.aov)

library(dplyr)

group_by(anovaFrame, df.religion1) %>%
  summarise(
    count = n(),
    mean = mean(Scale1, na.rm = TRUE),
    sd = sd(Scale1, na.rm = TRUE)
  )

ggplot(anovaFrame, aes(x = df.religion1, y = Scale1 )) +
  geom_boxplot(color="blue")+labs(x="Religion",y="Game score")
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## df.religion1  1    5376    5376   0.079   0.78
```

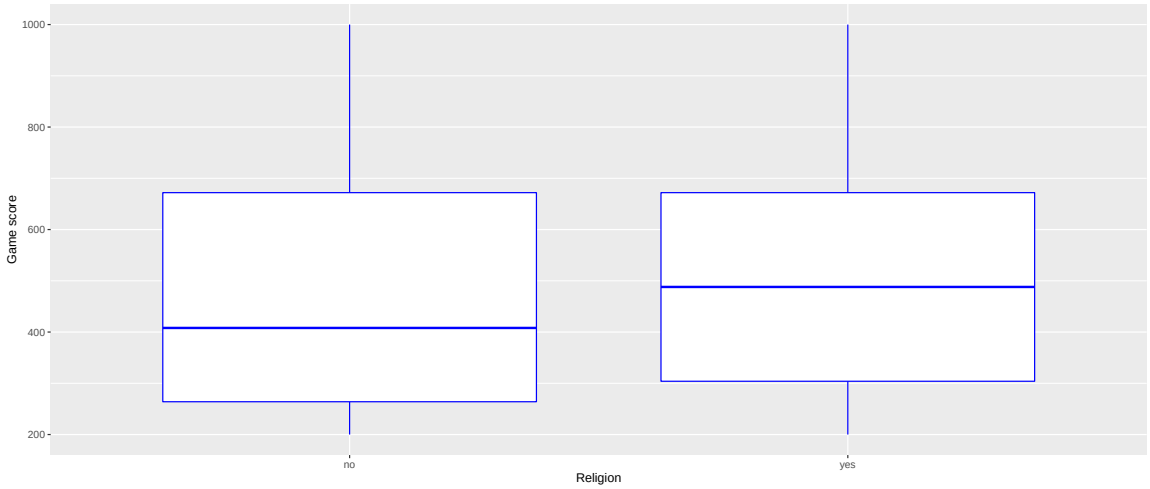

Figure SA1-13
Religion: Correlations with game score

```
## Residuals    98 6690304   68268
## # A tibble: 2 x 4
##   df.religion1 count  mean    sd
##   <fct>        <int> <dbl> <dbl>
## 1 no              70  485.  271.
## 2 yes             30  501.  237.
```

Religion.

```
anovaFrame=data.frame(
  party=df$polParty1,
  scoresTG
)

res.aov <- aov(Scale1 ~ party, data=anovaFrame)
summary(res.aov)

library(dplyr)

group_by(anovaFrame, party) %>%
  summarise(
    count = n(),
    mean = mean(Scale1, na.rm = TRUE),
    sd = sd(Scale1, na.rm = TRUE)
  )
```

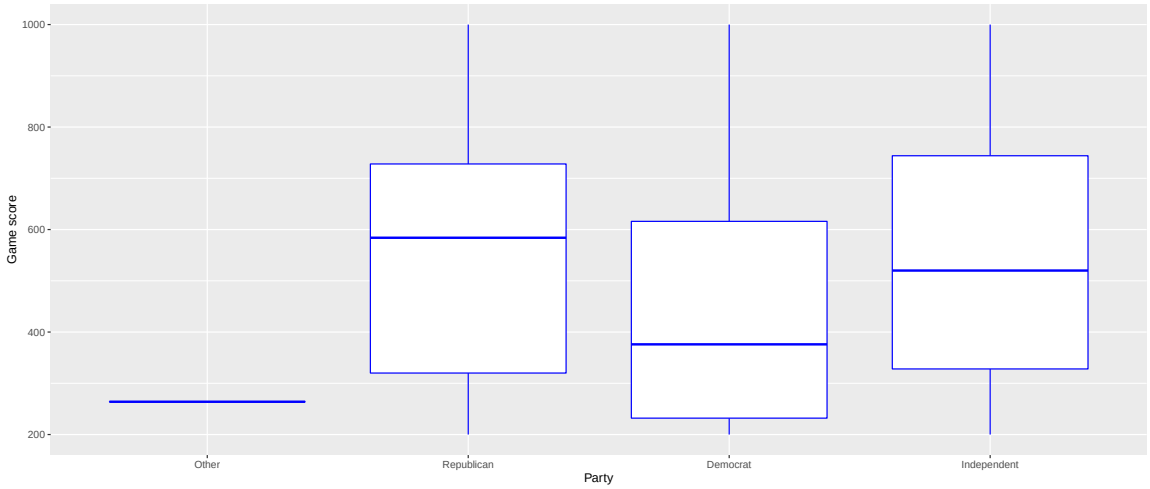

Figure SA1-14
Political party: Correlations with game score

```
ggplot(anovaFrame, aes(x = party, y = Scale1 )) +
  geom_boxplot(color="blue")+labs(x="Party",y="Game score")
```

```
##             Df  Sum Sq Mean Sq F value Pr(>F)
## party        3  334140  111380   1.681  0.176
## Residuals   96 6361540   66266
## # A tibble: 4 x 4
##   party       count  mean    sd
##   <fct>       <int> <dbl> <dbl>
## 1 Other           1  264    NA
## 2 Republican     16  566   278.
## 3 Democrat       52  442.  248.
## 4 Independent    31  538.  262.
```

Political party.

SA1-4 Exploratory analysis of relationships between variables

SA1-4.1 COVID-19 related scales and game scores

```
corData=data.frame(
  scoresCVW-1,
  scoresCVCP,
  scoresCVMI,
  scoresCVSD,
  scoresCVTO,
```

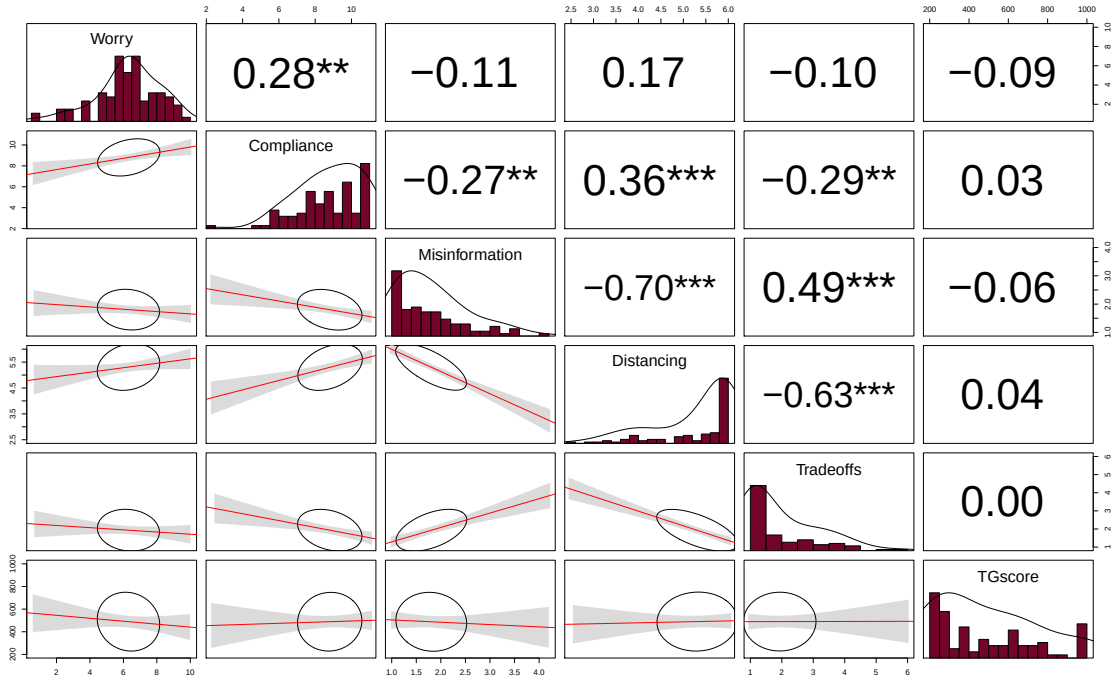

Figure SA1-15
COVID-19 scales: Correlations with game score

```
  scoresTG
)

corData <- corData %>%
  rename(
    Worry=sclCOVWorry,
    Compliance=sclCompliance,
    Misinformation=sclMisinform,
    Distancing=sclCOVSocDist,
    Tradeoffs=sclTradeoffs,
    TGscore=Scale1
  )

pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#73052b",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

SA1-4.2 COVID-19 related scales and HEXACO scales

```
 corData=data.frame(
   scoresHEXACO$scores,
   scoresCVW,
   scoresCVCP,
   scoresCVSD,
   scoresCVMI,
   scoresCVTO
 )



corData <- corData %>%
 rename(
VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs,
briefH=scaleBH,
briefE=scaleBE,
briefX=scaleBX,
briefA=scaleBA,
briefC=scaleBC,
briefO=scaleBO,
longH=scaleH60
 )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="blue",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

SA1-4.3 COVID-19 related scales and other personality scales

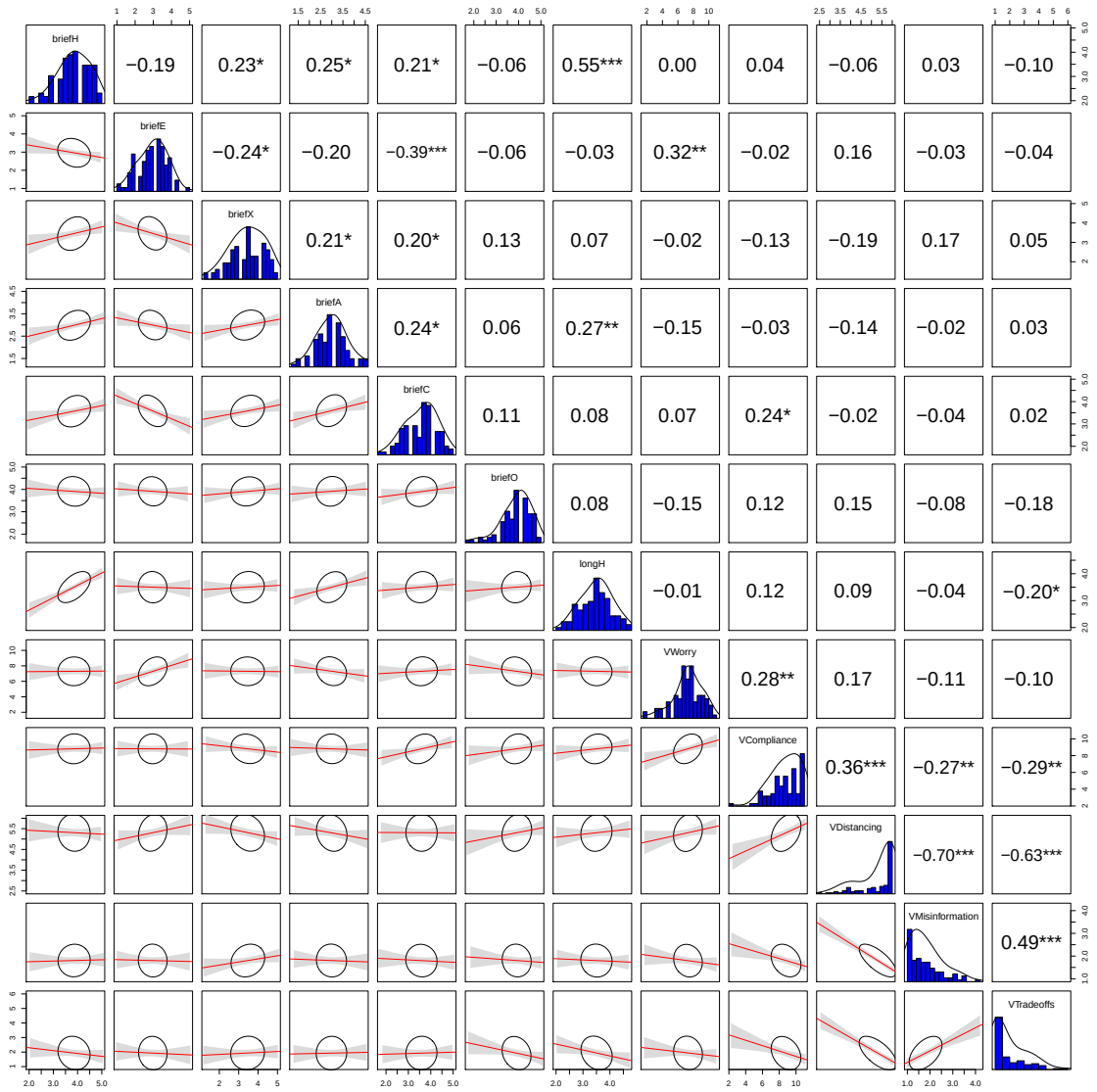

Figure SA1-16

HEXACO scales: Correlations with COVID-19 scales

```
corData=data.frame(
  scoresDC,
  scoresPR,
  scoresOUS,
 scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)


corData <- corData %>%
  rename(
    DarkCore=scaleDC,
    PsychReactance=scalePR,
    ImpBeneficence=scaleOUIB,
    InstrHarm=scaleOUIH,
VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs
)


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#4b7a42",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

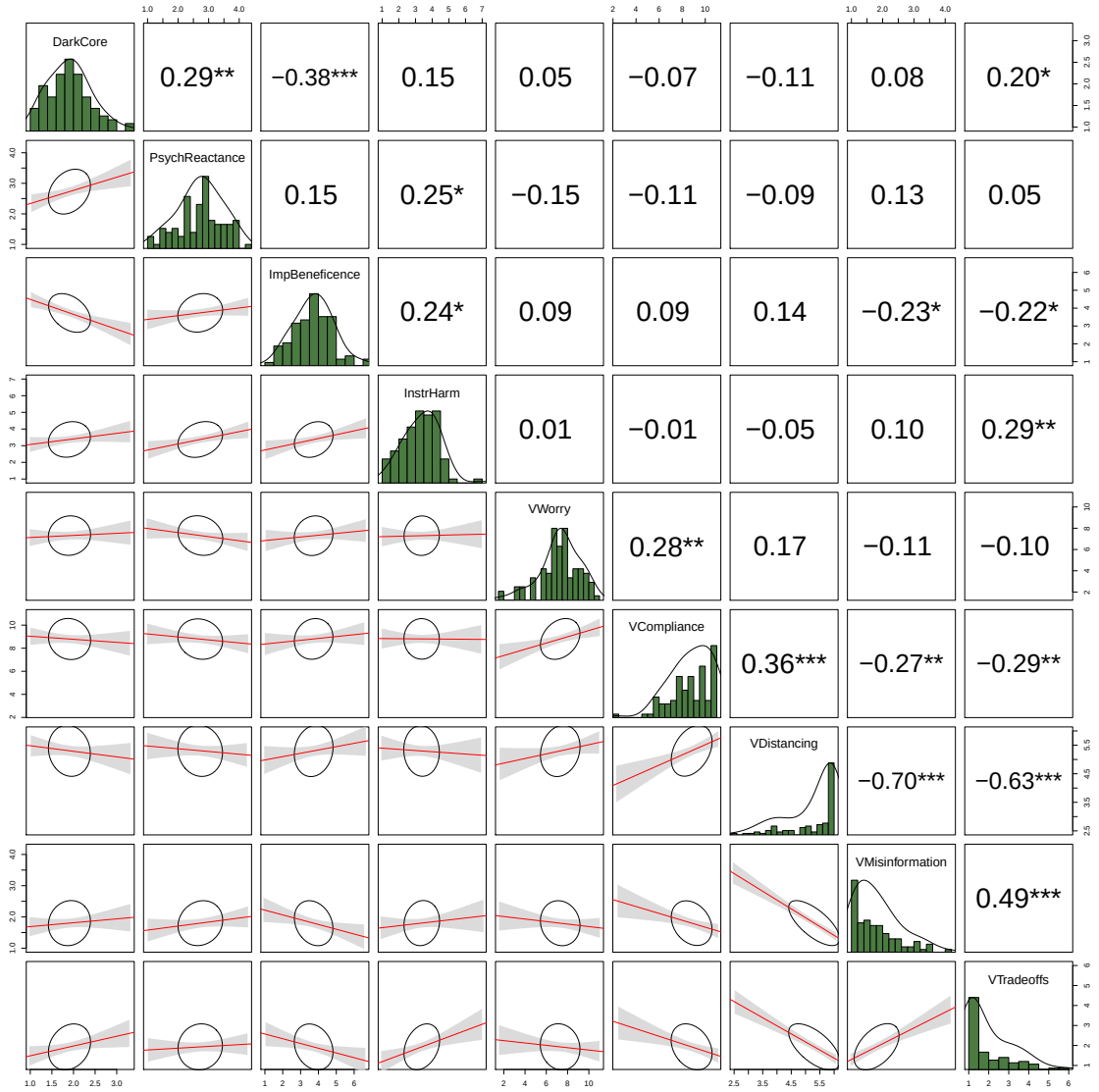

Figure SA1-17

Other personality scales: Correlations with COVID-19 scales

SA1-4.4 COVID-19 related scales and cognitive variables

```
corData=data.frame(
  scoresCRT,
  scoresBNT,
  scoresEXP,
  scoresSN,
 scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)


corData <- corData %>%
  rename(
    CRT=CRTscore,
    CRTi=CRTintuitive,
    BNT=BNTscorBNT,
    EXPw=EXPscoreWide,
    EXPn=EXPscoreNarrow,
    SN=Scale1,
VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs,
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#30a18e",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

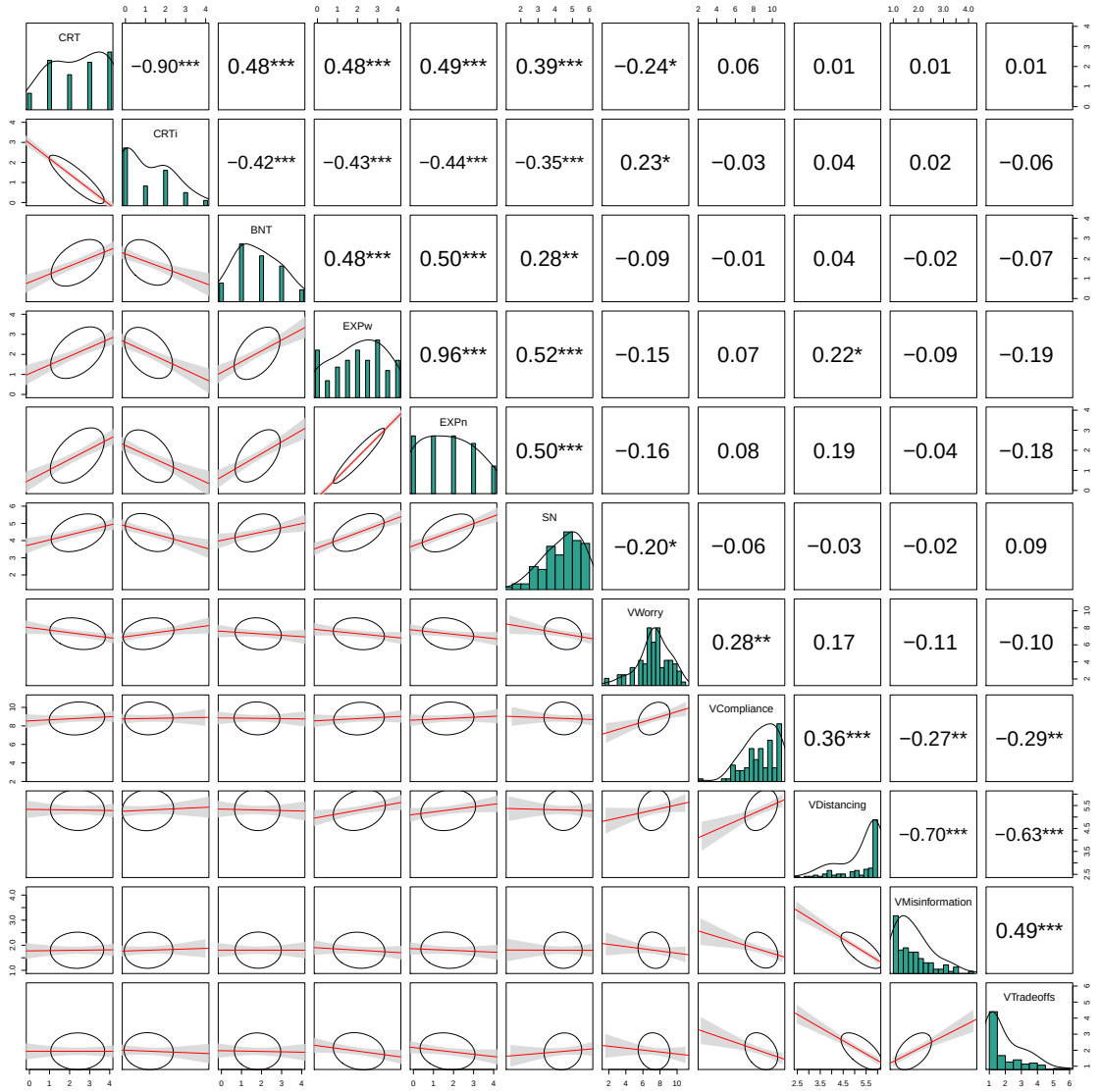

Figure SA1-18
Cognitive variables: Correlations with COVID-19 scales

SA1-4.5 COVID-19 related scales and economic games

```
corData=data.frame(
  df$DictatorGame,
  df$DictatorExpect,
  df$MonBurnGame,
  df$MonBurnExpect,
  scoresDICE,
  angleSVO,
 scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)



corData <- corData %>%
  rename(
    DictatorA=df.DictatorGame,
    DictatorE=df.DictatorExpect,
    MonBurnA=df.MonBurnGame,
    MonBurnE=df.MonBurnExpect,
    Dicescore=Scale1,
    SVO=angleSVO,
VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs
  )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#825503",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

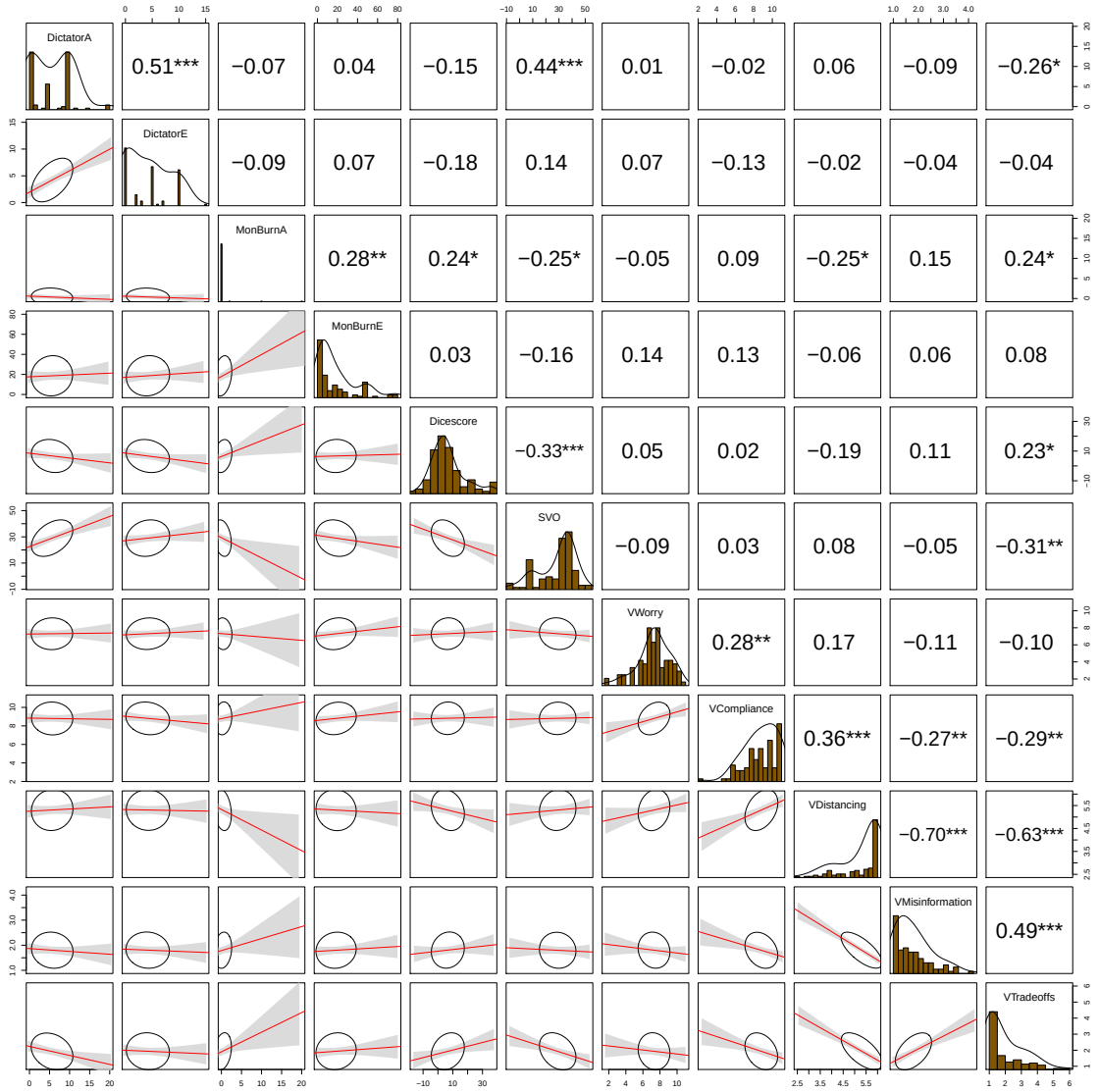

Figure SA1-19

Economic Games: Correlations with COVID-19 related scales

SA1-4.6 COVID-19 related scales and single-item personality measures

```
# variables are adjusted to R's conversion defaults

corData=data.frame(
  as.numeric(df$RTGeneral)-1,
  as.numeric(df$TRUST)-1,
  as.numeric(df$TIMEPREFERENCE)-1,
  as.numeric(df$IMPATIENCE)-1,
  as.numeric(df$IMPULSIVENESS)-1,
   scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)



# variable names are adjusted to R's variable naming defaults....

corData <- corData %>%
  rename(
    RiskPreference=as.numeric.df.RTGeneral....1,
    Trust=as.numeric.df.TRUST....1,
    TimePreference=as.numeric.df.TIMEPREFERENCE....1,
    Impatience=as.numeric.df.IMPATIENCE....1,
    Impulsiveness=as.numeric.df.IMPULSIVENESS....1,
    VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs
  )


pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#1f0b52",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

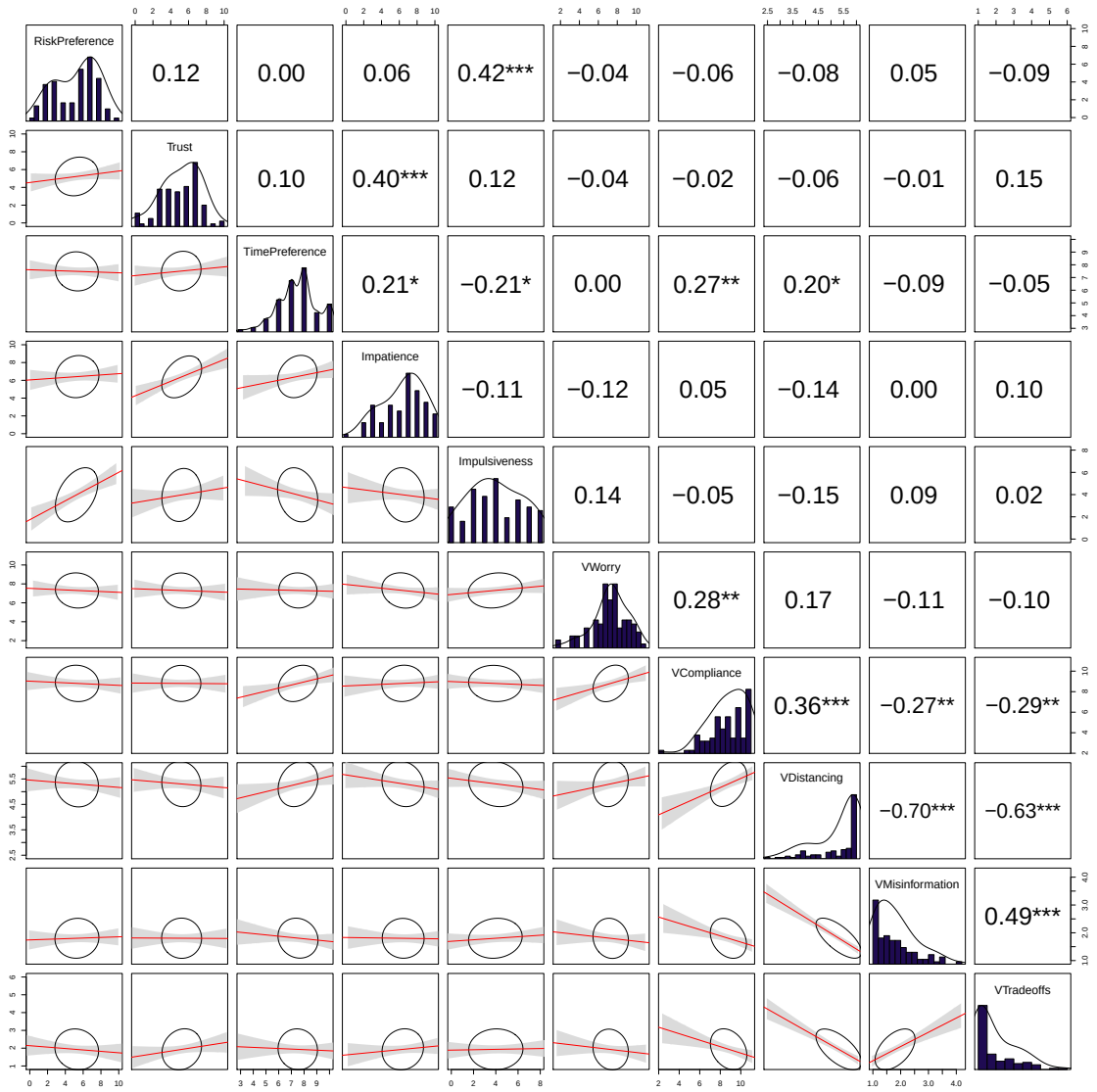

Figure SA1-20

Single-item personality variables: Correlations with COVID-19 related scales

SA1-4.7 COVID-19 related scales and political measures

```
corData=data.frame(
 as.numeric(df$polPosition),
 scoresCandidates,
 scoresCC,
 scoresSECS,
  scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)


corData <- corData %>%
  rename(
    Conservatism=as.numeric.df.polPosition.,
    PrefTrump=Trump,
    PrefBiden=Biden,
    ClimateSkepticism=sclClimSkept,
    SECS=sclConsALL,
    SECSsocial=sclConsSoc,
    SECSeconomic=sclConsEcon,
    VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs
  )

pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#7f07b3",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

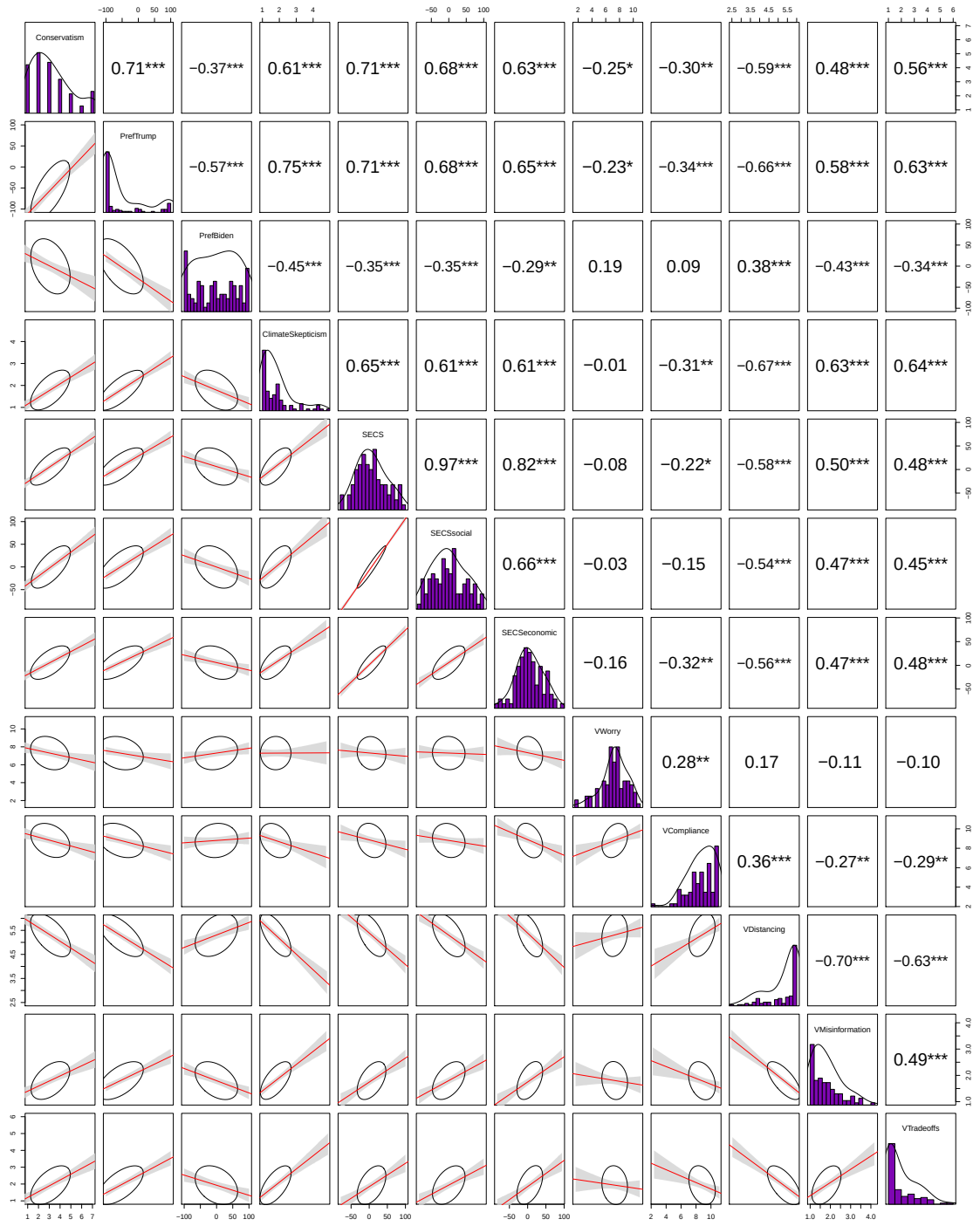

Figure SA1-21

Political measures: Correlations with COVID-19 related scores

SA1-4.8 COVID-19 related scales and age and health

```
corData=data.frame(
  df$demo01Age,
  df$GenHealth,
  df$SLExp_1,
  SubjYearsLeft=df$SLExp_1-df$demo01Age,
   scoresCVW,
    scoresCVCP,
    scoresCVSD,
    scoresCVMI,
    scoresCVTO
)



corData <- corData %>%
  rename(
    Age=df.demo01Age,
    GeneralHealth=df.GenHealth,
    SubjLifeExpect=df.SLExp_1,
    VWorry=sclCOVWorry,
VCompliance=sclCompliance,
VDistancing=sclCOVSocDist,
VMisinformation=sclMisinform,
VTradeoffs=sclTradeoffs
  )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#990612",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

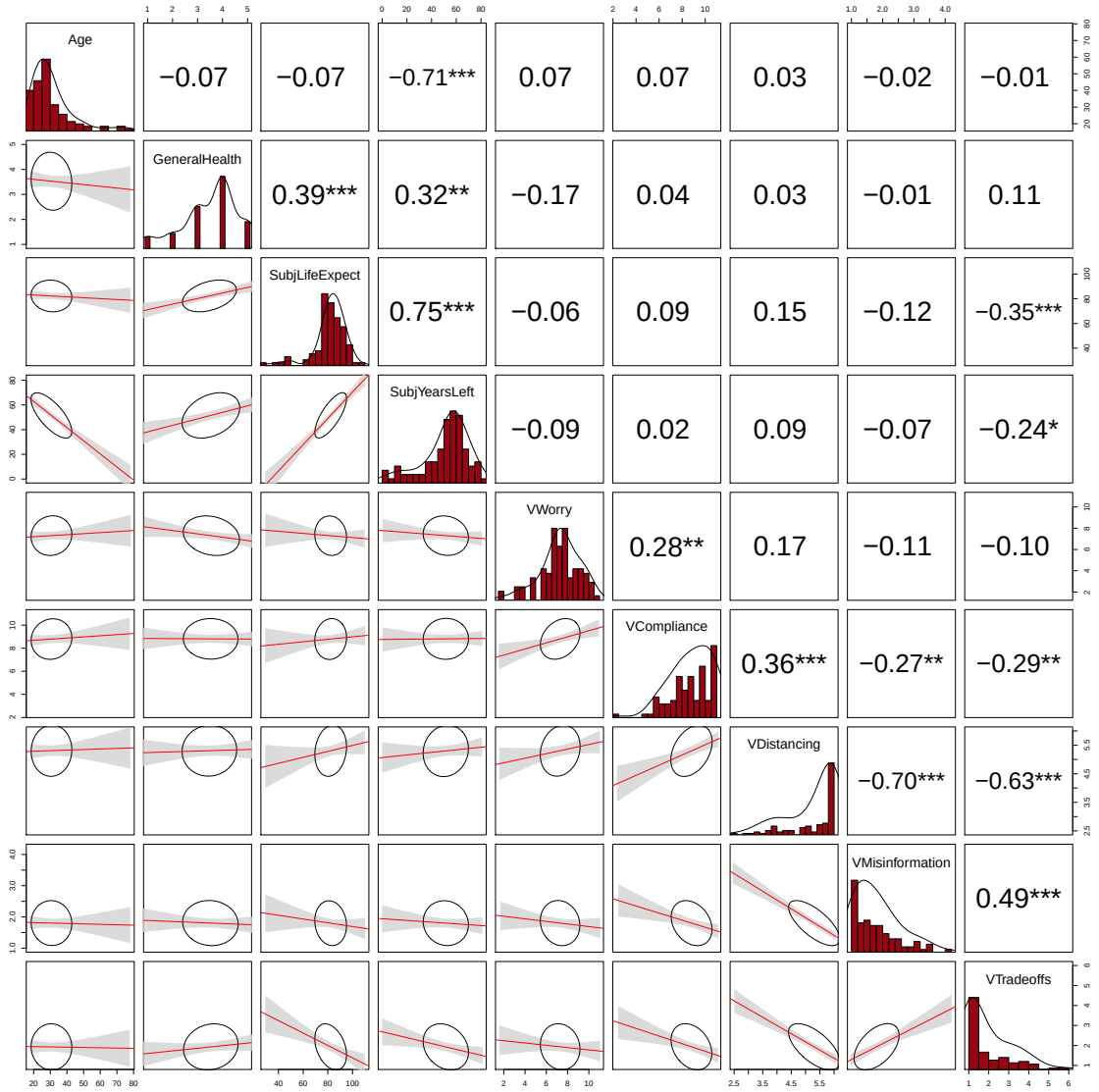

Figure SA1-22

Age and health: Correlations with COVID-19 related scales

SA1-4.9 Multiple regression models: Predicting game scores

```
  aovData=data.frame(
    scoresHEXACO$scores,
    scoresCRT,
    as.numeric(df$RTGeneral)-1,
    angleSVO,
     scoresTG
  )

aovData <- aovData %>%
  rename(
   HEXH=scaleH60,
   HEXE=scaleBE,
   Risk=as.numeric.df.RTGeneral....1,
   CRT=CRTscore,
   SVO=angleSVO,
   GameRes=Scale1
  )

Reg_model <- lm(formula= GameRes ~ CRT + HEXH+HEXE+Risk+SVO, data = aovData)

summary(Reg_model)

##
## Call:
## lm(formula = GameRes ~ CRT + HEXH + HEXE + Risk + SVO, data = aovData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -500.29 -181.21  -41.23  180.24  514.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1113.152    211.689   5.258 9.08e-07 ***
## CRT          -47.508     18.211  -2.609  0.01057 *
## HEXH         -72.179     46.279  -1.560  0.12220
## HEXE         -95.475     31.997  -2.984  0.00363 **
## Risk          11.353     10.521   1.079  0.28332
## SVO           -1.190      1.926  -0.618  0.53814
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 235.3 on 94 degrees of freedom
## Multiple R-squared:  0.2228,Adjusted R-squared:  0.1815
```

```
## F-statistic: 5.389 on 5 and 94 DF,  p-value: 0.0002113
```

SA1-4.9.1 Motivation

Five questions were answered on a scale from 1–5: Strongly disagree—Disagree—Neither agree nor disagree—Agree—Strongly agree.

- **Post1**: I wanted to make as much bonus money as possible.
- **Post2**: I wanted to make more bonus money than other players.
- **Post3**: I felt responsible for other players.
- **Post4**: I wanted to make other players switch color.
- **Post5**: I did not care at all what happened in this game.

```
  corData=data.frame(
    df$PostTG_1,
    df$PostTG_2,
    df$PostTG_3,
    df$PostTG_4,
    df$PostTG_5,
    scoresTG
  )
```

```
corData <- corData %>%
  rename(
    MaxBonus=df.PostTG_1,
    MoreBonus=df.PostTG_2,
    Responsible=df.PostTG_3,
    MakeSwitch=df.PostTG_4,
    DoNotCare=df.PostTG_5,
    TGscore=Scale1
  )
```

```
pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#5c5706",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

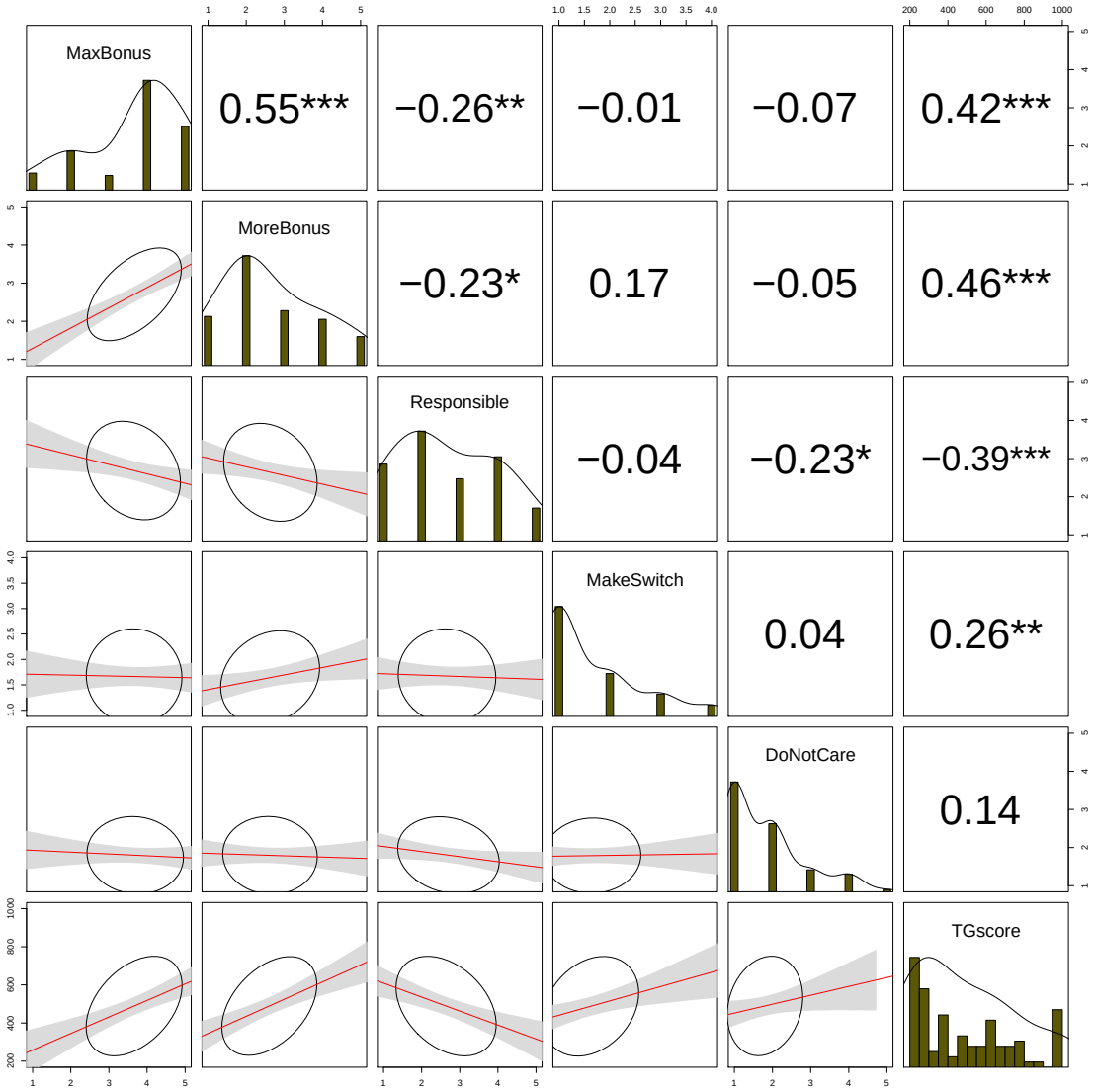

Figure SA1-23
Postquestionnaire motivation: Correlation with game score

SA1-4.9.2 Expectation Purple

```
    corData=data.frame(
    df$PostTGestimatepurple_1,
    df$PostTGestimatepurple_2,
    df$PostTGestimatepurple_3,
    df$PostTGestimatepurple_4,
    df$PostTGestimatepurple_5,
    df$PostTGestimatepurple_6,
    df$PostTGcolor,
    scoresTG
  )



corData <- corData %>%
  rename(
    PurpleR01=df.PostTGestimatepurple_1,
    PurpleR05=df.PostTGestimatepurple_2,
    PurpleR10=df.PostTGestimatepurple_3,
    PurpleR15=df.PostTGestimatepurple_4,
    PurpleR20=df.PostTGestimatepurple_5,
    PurpleR25=df.PostTGestimatepurple_6,
    SelfPurple=df.PostTGcolor,
    TGscore=Scale1
  )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#440563",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

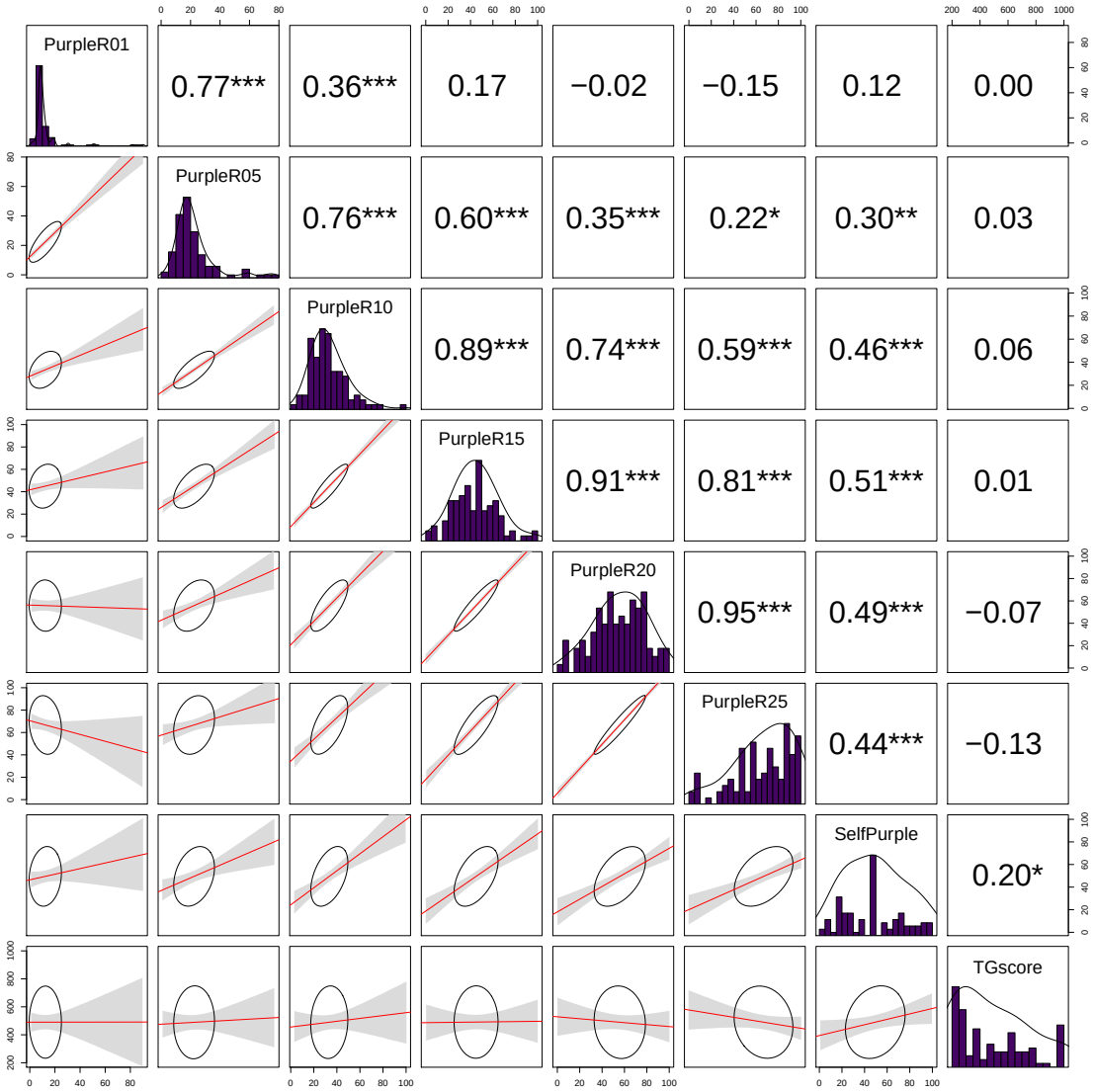

Figure SA1-24

Postquestionnaire expectation percent purple: Correlation with game score

SA1-4.9.3 Percent H-choices

```
    corData=data.frame(
    df$PostTGestimateH_1,
    df$PostTGestimateH_2,
    df$PostTGestimateH_3,
    df$PostTGestimateH_4,
    df$PostTGestimateH_5,
    df$PostTGestimateH_6,
    df$PostTGcolor,
    scoresTG
  )



corData <- corData %>%
  rename(
    HchosenR01=df.PostTGestimateH_1,
    HchosenR05=df.PostTGestimateH_2,
    HchosenR10=df.PostTGestimateH_3,
    HchosenR15=df.PostTGestimateH_4,
    HchosenR20=df.PostTGestimateH_5,
    Hchosen25=df.PostTGestimateH_6,
    SelfPurple=df.PostTGcolor,
    TGscore=Scale1
  )



pairs.panels(corData, smooth = TRUE, scale = FALSE,  digits = 2,
             method="pearson",pch = 20, lm=TRUE,cor=TRUE,jiggle=TRUE,
             factor=2,breaks=15,
             hist.col="#440563",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

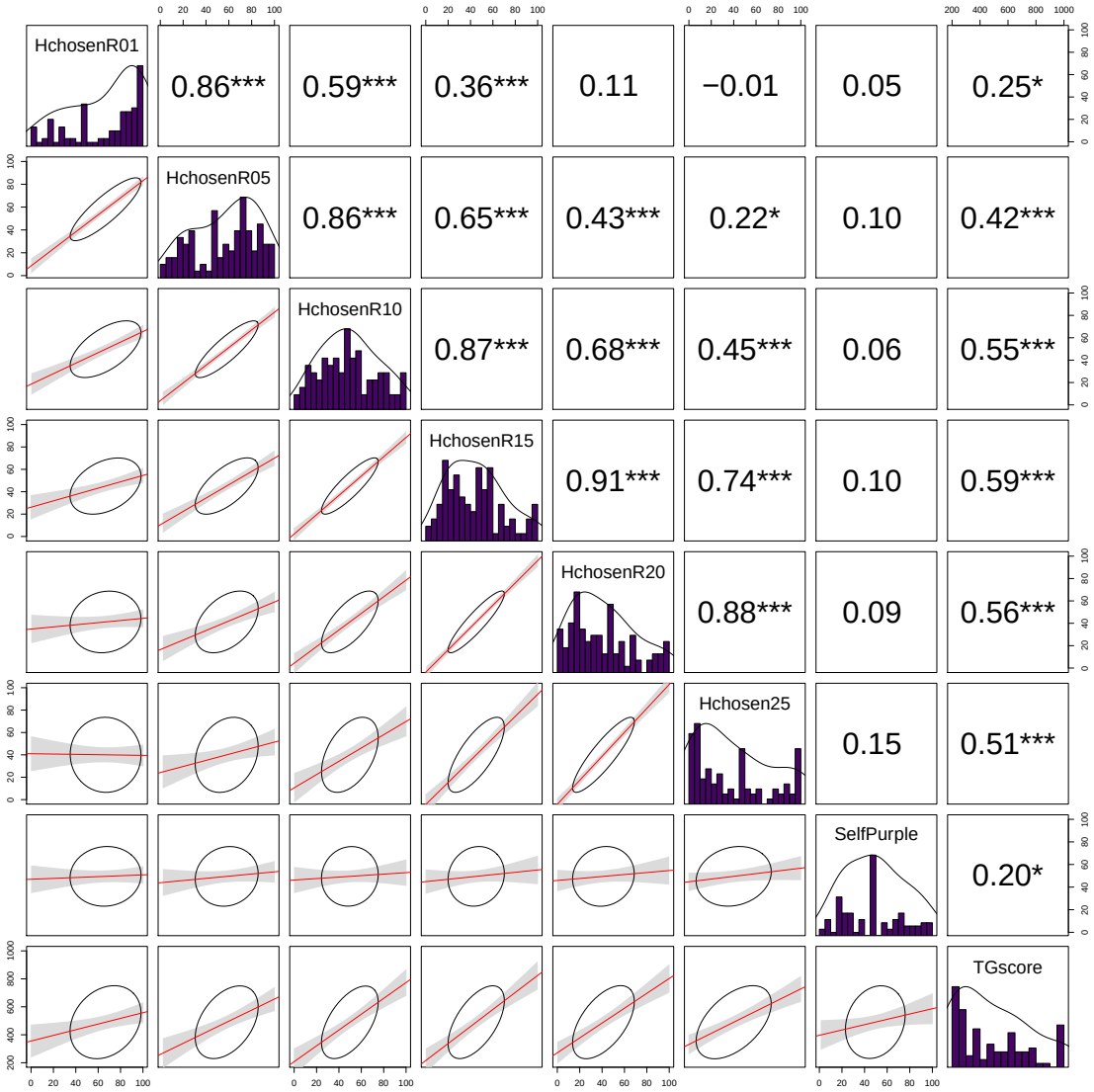

Figure SA1-25

Postquestionnaire expectation percent H-choices: Correlation with game score