October 24, 2021

Transmission Game: Supporting Analyses Study 2 (SA2)

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 2 (SA2)

Contents

SA2-1 Data preparation **3**
SA2-1.1 Packages and dataset 3
SA2-1.2 Variables and scale construction 6

SA2-2 Exclusion of participants and sample selection **19**

SA2-3 Transmission game variables **20**
SA2-3.1 Calculating scales 20
SA2-3.2 Classifying behavioral patterns 29
SA2-3.3 Behavioral patterns and conditions 31
SA2-3.4 Behavioral pattern and risk preference 34
SA2-3.5 Behavioral pattern and CRT 35

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

SA2-3.6 Behavioral pattern and SVO 36
SA2-3.7 Behavioral pattern and gender 38

SA2-4 Game results and predictors 40
SA2-4.1 Technical note 40
SA2-4.2 Predictors identified in Study 1 42
SA2-4.3 Exploratory analysis 42
SA2-4.4 HEXACO-scales 42
SA2-4.5 Other personality scales 43
SA2-4.6 Cognitive variables 45
SA2-4.7 Economic games 45
SA2-4.8 Political measures 47
SA2-4.9 Categorical variables 48
SA2-4.9.1 Demographics 49
SA2-4.9.2 Religion and politics 53

SA2-5 Game behavior and postquestionnaire 56
SA2-5.1 Motivation 56
SA2-5.2 Motivation and risk preference 59
SA2-5.3 Motivation and SVO 62
SA2-5.4 Motivation and CRT 63
SA2-5.5 Specific postquestionnaires and game behavior 65
SA2-5.6 Expectation Purple 67
SA2-5.6.1 Percent H-choices 76

SA2-6 Exploratory Analyses 85
SA2-6.1 COVID-19 related scales and predictor variables 85
SA2-6.2 COVID-19 related scales and HEXACO-scales 86
SA2-6.3 COVID-19 related scales and politics 86
SA2-6.4 COVID-19 testing and game results 95

List of Figures

SA2-1 Transmission Game results distribution 23
SA2-2 Transmission Game roundwise results 25
SA2-3 Game results across conditions: Jittered points within the box-plots represent individual results, p-values of selected comparisons are shown. 27
SA2-4 Experimental conditions and distribution of percentage of risky choices 29
SA2-5 Experimental conditions and behavioral patterns 33
SA2-6 Behavioral pattern and risk preference 36
SA2-7 Behavioral pattern and CRT 37
SA2-8 Behavioral pattern and SVO 39
SA2-9 Behavioral pattern and gender 40
SA2-10 Game and HEXACO-scales 43
SA2-11 Game and personality 44

SA2-12 Game and CRT . . . 46
SA2-13 Game and SVO . . . 47
SA2-14 Game and politics . . . 49
SA2-15 Gender and game scores . . . 51
SA2-16 Education and game scores . . . 52
SA2-17 HH income and game scores . . . 53
SA2-18 Religion and game scores . . . 55
SA2-19 Political party and game scores . . . 56
SA2-20 Game and Poastquestionnaire (part 1) . . . 58
SA2-21 Game and Poastquestionnaire (part 2) . . . 60
SA2-22 Motivation and risk preference . . . 61
SA2-23 Motivation and SVO . . . 63
SA2-24 Motivation and CRT . . . 65
SA2-25 Reactions to interventions and game scores . . . 67
SA2-26 Postquestionnaire expectation percent purple . . . 69
SA2-27 Postquestionnaire expectation percent purple . . . 72
SA2-28 Postquestionnaire expectation percent H-choices . . . 78
SA2-29 Postquestionnaire expectation percent H-choices . . . 80
SA2-30 COVID-19 related scales and predictors . . . 88
SA2-31 COVID-19 related scales and HEXACO scales . . . 89
SA2-32 COVID-19 related scales and politics . . . 90
SA2-33 Conservatism and Compliance . . . 92
SA2-34 Preference for Trump and Compliance . . . 92
SA2-35 Preference for Biden and Compliance . . . 93
SA2-36 Conservatism and Tradeoffs . . . 93
SA2-37 Preference for Trump and Treadeoffs . . . 94
SA2-38 Conservatism and Worries . . . 94
SA2-39 COVID-19 testing and game results . . . 96

List of Tables

SA2-1 Error count frequencies in comprenhension check . . . 21

SA2-1 Data preparation

SA2-1.1 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)
```

```r
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_STUDY2_DEIDENTIFIED.sav'
dataS=read.spss(file=fn)
```

```
## Warning in read.spss(file = fn):  TRANSMISSION_GAME_STUDY2_DEIDENTIFIED.sav:
Very long string record(s) found (record type 7, subtype 14), each will be
imported in consecutive separate variables
```

```
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
```

SA2-1.2 Variables and scale construction

```
    VarsPost <- c("PostTG_01", "PostTG_02", "PostTG_03",
    "PostTG_04", "PostTG_05","PostTG_06","PostTG_07",
    "PostTG_08","PostTG1_09","PostTG_10","PostTG_11")



    FramePost <- df[VarsPost]

  FramePost <- FramePost %>%
    rename(
    Post01=PostTG_01,Post02=PostTG_02,Post03=PostTG_03,Post04=PostTG_04,
    Post05=PostTG_05,Post06=PostTG_06,Post07=PostTG_07,Post08=PostTG_08,
    Post09=PostTG1_09,Post10=PostTG_10,Post11=PostTG_11
    )

    FramePost[] <-data.matrix(FramePost)
```

```
#Data preparation:
```

```
round<-c()
round[1:600]<-1
round[601:1200]<-5
round[1201:1800]<-10
round[1801:2400]<-15
round[2401:3000]<-20
round[3001:3600]<-25

participantID<-c()
participantID[1:600]<-1:600
participantID[601:1200]<-1:600
participantID[1201:1800]<-1:600
participantID[1801:2400]<-1:600
participantID[2401:3000]<-1:600
participantID[3001:3600]<-1:600

estimatePurple<-c()
estimatePurple[1:600] <-df$PostTGestimatepurple_1
estimatePurple[601:1200] <-df$PostTGestimatepurple_2
estimatePurple[1201:1800] <-df$PostTGestimatepurple_3
estimatePurple[1801:2400] <-df$PostTGestimatepurple_4
estimatePurple[2401:3000] <-df$PostTGestimatepurple_5
estimatePurple[3001:3600] <-df$PostTGestimatepurple_6

estimateH<-c()
estimateH[1:600] <-df$PostTGestimateH_1
estimateH[601:1200] <-df$PostTGestimateH_2
estimateH[1201:1800] <-df$PostTGestimateH_3
estimateH[1801:2400] <-df$PostTGestimateH_4
estimateH[2401:3000] <-df$PostTGestimateH_5
estimateH[3001:3600] <-df$PostTGestimateH_6

lineFramePurple<-data.frame(
  round,
  estimatePurple,
  participantID
)

lineFrameH<-data.frame(
  round,
  estimateH,
  participantID
)
```

```
  VarsPostCR <- c("AddPostquestChain_1", "AddPostquestChain_2",
  "AddPostquestChain_3", "AddPostquestChain_4",
  "AddPostquestChain_5","AddPostquestChain_6")

  FramePostCR <- df[VarsPostCR]

FramePostCR <- FramePostCR %>%
  rename(
  PostCR1=AddPostquestChain_1,PostCR2=AddPostquestChain_2,
  PostCR3=AddPostquestChain_3,PostCR4=AddPostquestChain_4,
  PostCR5=AddPostquestChain_5,PostCR6=AddPostquestChain_6
  )

  FramePostCR[] <-data.matrix(FramePostCR)
```

```
  VarsPostSIM <- c("AddPostquestSimul_1", "AddPostquestSimul_2",
  "AddPostquestSimul_3", "AddPostquestSimul_4",
  "AddPostquestSimul_5","AddPostquestSimul_6","AddPostquestSimul_7")

  FramePostSIM <- df[VarsPostSIM]

FramePostSIM <- FramePostSIM %>%
  rename(
  PostSIM1=AddPostquestSimul_1,PostSIM2=AddPostquestSimul_2,
  PostSIM3=AddPostquestSimul_3,PostSIM4=AddPostquestSimul_4,
  PostSIM5=AddPostquestSimul_5,PostSIM6=AddPostquestSimul_6,
  PostSIM7=AddPostquestSimul_7
  )

  FramePostSIM[] <-data.matrix(FramePostSIM)
```

```
  VarsPostVL <- c("AddPostVicar_1", "AddPostVicar_2",
  "AddPostVicar_3", "AddPostVicar_4",
  "AddPostVicar_5","AddPostVicar_6")

  FramePostVL <- df[VarsPostVL]

FramePostVL <- FramePostVL %>%
  rename(
  PostVL1=AddPostVicar_1,PostVL2=AddPostVicar_2,
  PostVL3=AddPostVicar_3,PostVL4=AddPostVicar_4,
  PostVL5=AddPostVicar_5,PostVL6=AddPostVicar_6
  )
```

```
   FramePostVL[] <-data.matrix(FramePostVL)
```

```
   VarsPostDN <- c("AddPostDescriptive_1", "AddPostDescriptive_2",
   "AddPostDescriptive_3", "AddPostDescriptive_4",
   "AddPostDescriptive_5","AddPostDescriptive_6")

   FramePostDN <- df[VarsPostDN]

 FramePostDN <- FramePostDN %>%
   rename(
   PostDN1=AddPostDescriptive_1,PostDN2=AddPostDescriptive_2,
   PostDN3=AddPostDescriptive_3,PostDN4=AddPostDescriptive_4,
   PostDN5=AddPostDescriptive_5,PostDN6=AddPostDescriptive_6
   )

   FramePostDN[] <-data.matrix(FramePostDN)
```

```
   VarsPostIN <- c("AddPostInjunctive_1", "AddPostInjunctive_2",
   "AddPostInjunctive_3", "AddPostInjunctive_4",
   "AddPostInjunctive_5","AddPostInjunctive_6")

   FramePostIN <- df[VarsPostIN]

 FramePostIN <- FramePostIN %>%
   rename(
   PostIN1=AddPostInjunctive_1,PostIN2=AddPostInjunctive_2,
   PostIN3=AddPostInjunctive_3,PostIN4=AddPostInjunctive_4,
   PostIN5=AddPostInjunctive_5,PostIN6=AddPostInjunctive_6
   )

   FramePostIN[] <-data.matrix(FramePostIN)
```

```
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\nconser-\nvative", after = 4)
```

```
 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)

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

```
 scaleVars <- c("HexBHI_O01", "HexBHI_C02", "HexBHI_A03R",
"HexBHI_X04R","HexBHI_O07R","HexBHI_C08R","HexBHI_A09R",
"HexBHI_X10","HexBHI_O13","HexBHI_C14","HexBHI_A15",
"HexBHI_X16","HexBHI_O19","HexBHI_C20R","HexBHI_A21",
"HexBHI_X22R", "HexacoH_06", "HexacoH_12R", "HexacoH_18",
"HexacoH_24R", "HexacoH_30R", "HexacoH_36", "HexacoH_42R",
"HexacoH_48R", "HexacoH_54", "HexacoH_60R","HexacoE_05",
"HexacoE_11","HexacoE_17","HexacoE_23","HexacoE_29",
"HexacoE_35R","HexacoE_41R","HexacoE_47","HexacoE_53R",
"HexacoE_59R" )


 scaleFrame <- df[scaleVars]

 scaleFrame <- scaleFrame %>%
  rename(
  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,
  E60_01p=HexacoE_05,E60_02p=HexacoE_11,E60_03p=HexacoE_17,
  E60_04p=HexacoE_23,E60_05p=HexacoE_29,E60_06n=HexacoE_35R,
  E60_07n=HexacoE_41R,E60_08p=HexacoE_47,E60_09n=HexacoE_53R,
```

```
  E60_10n=HexacoE_59R,
  bO1p=HexBHI_O01, bC1p=HexBHI_C02, bA1n=HexBHI_A03R,
  bX1n=HexBHI_X04R,bO2n=HexBHI_O07R,bC2n=HexBHI_C08R,
  bA2n=HexBHI_A09R,bX2p=HexBHI_X10,bO3p=HexBHI_O13,
  bC3p=HexBHI_C14, bA3p=HexBHI_A15,bX3p=HexBHI_X16,
  bO4p=HexBHI_O19, bC4n=HexBHI_C20R,bA4p=HexBHI_A21,
  bX4n=HexBHI_X22R
)

 scaleFrame[] <-data.matrix(scaleFrame)

 weightsHEX <-list(
scaleH60= c("H60_01p","-H60_02n","H60_03p","-H60_04n",
            "-H60_05n","H60_06p","-H60_07n","-H60_08n",
            "H60_09p","-H60_10n"),
scaleE60= c("E60_01p","E60_02p","E60_03p","E60_04p",
            "E60_05p","-E60_06n","-E60_07n","E60_08p",
            "-E60_09n","-E60_10n"),
scaleBX = c("-bX1n","bX2p","bX3p","-bX4n"),
scaleBA =  c("-bA1n","-bA2n","bA3p","bA4p"),
scaleBC =  c("bC1p","-bC2n","bC3p","-bC4n"),
scaleBO = c("bO1p","-bO2n","bO3p","bO4p")
 )


 scaleVarsH60 <- c("H60_01p","H60_02n","H60_03p","H60_04n",
            "H60_05n","H60_06p","H60_07n","H60_08n",
            "H60_09p","H60_10n")

 scaleVarsE60= c("E60_01p","E60_02p","E60_03p","E60_04p",
            "E60_05p","E60_06n","E60_07n","E60_08p",
            "E60_09n","E60_10n")

scaleVarsBX <- c("bX1n","bX2p","bX3p","bX4n")
scaleVarsBA <- c("bA1n","bA2n","bA3p","bA4p")
scaleVarsBC <- c("bC1p","bC2n","bC3p","bC4n")
scaleVarsBO <- c("bO1p","bO2n","bO3p","bO4p")


scaleFrameH60 <- scaleFrame[scaleVarsH60]
scaleFrameE60 <- scaleFrame[scaleVarsE60]
scaleFrameBX <- scaleFrame[scaleVarsBX]
scaleFrameBA <- scaleFrame[scaleVarsBA]
scaleFrameBC <- scaleFrame[scaleVarsBC]
```

```
scaleFrameBO <- scaleFrame[scaleVarsBO]


scaleFrameH60[] <-data.matrix(scaleFrameH60)
scaleFrameE60[] <-data.matrix(scaleFrameE60)
scaleFrameBX[] <-data.matrix(scaleFrameBX)
scaleFrameBA[] <-data.matrix(scaleFrameBA)
scaleFrameBC[] <-data.matrix(scaleFrameBC)
scaleFrameBO[] <-data.matrix(scaleFrameBO)

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

```
   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)
head(scoresCRT)
```

```
##   CRTscore CRTintuitive
## 1        4            0
## 2        3            1
## 3        1            2
## 4        3            1
## 5        0            2
## 6        1            3
```

```
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))
```

```
    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)


summary(scaleFrameSVO)

##     SVO1self    SVO1other        SVO2self        SVO2other        SVO3self
##  Min.   :85   Min.   :15.00   Min.   : 85.00   Min.   :15.00   Min.   :50.00
##  1st Qu.:85   1st Qu.:85.00   1st Qu.:100.00   1st Qu.:50.00   1st Qu.:85.00
##  Median :85   Median :85.00   Median :100.00   Median :50.00   Median :85.00
##  Mean   :85   Mean   :79.95   Mean   : 99.33   Mean   :48.46   Mean   :83.13
##  3rd Qu.:85   3rd Qu.:85.00   3rd Qu.:100.00   3rd Qu.:50.00   3rd Qu.:85.00
##  Max.   :85   Max.   :85.00   Max.   :100.00   Max.   :50.00   Max.   :85.00
##    SVO3other         SVO4self       SVO4other         SVO5self
##  Min.   : 85.00   Min.   :50.00   Min.   : 15.00   Min.   : 50.00
##  1st Qu.: 85.00   1st Qu.:63.00   1st Qu.: 36.00   1st Qu.: 75.00
##  Median : 85.00   Median :68.00   Median : 58.00   Median : 75.00
##  Mean   : 85.81   Mean   :69.18   Mean   : 53.85   Mean   : 82.79
##  3rd Qu.: 85.00   3rd Qu.:76.00   3rd Qu.: 68.00   3rd Qu.:100.00
##  Max.   :100.00   Max.   :85.00   Max.   :100.00   Max.   :100.00
##    SVO5other         SVO6self       SVO6other
##  Min.   : 50.00   Min.   : 85.0   Min.   :50.00
##  1st Qu.: 50.00   1st Qu.: 85.0   1st Qu.:59.00
##  Median : 75.00   Median : 85.0   Median :85.00
##  Mean   : 67.25   Mean   : 89.5   Mean   :74.58
##  3rd Qu.: 75.00   3rd Qu.: 96.0   3rd Qu.:85.00
##  Max.   :100.00   Max.   :100.0   Max.   :85.00

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

 weightsSVO <-list(sclSVOself=c("SVO2self","SVO3self",
         "SVO4self","SVO5self","SVO6self"),
     sclSVOother=c("SVO1other","SVO2other","SVO3other","SVO4other",
```

```
         "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)

  head(scoresSVO)

##   angleSVO
## 1 36.44444
## 2 23.83874
## 3 29.43257
## 4 18.14558
## 5 29.13841
## 6 36.44444
```

```
    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.
```

```
 #The summary scores are translated back to the original 0-10 interval
#(R interprets the original numbers as factor levels)

 scoresCVW<-data.frame(scaleCVW$scores-1)
```

```
     scaleVarsCVCP <- c("COV_guidelines", "COV_mask",
     "COV_meet", "COV_vaccine","Q579" )

    scaleFrameCVCP <- df[scaleVarsCVCP]

  scaleFrameCVCP <- scaleFrameCVCP %>%
    rename(
    CVCP1p=COV_guidelines,CVCP2p=COV_mask,CVCP3n=COV_meet,
    CVCP4p=COV_vaccine, CVCP5n=Q579
    )

    scaleFrameCVCP[] <-data.matrix(scaleFrameCVCP)

 weightsCVCP <-list(sclCompliance=c("CVCP1p","CVCP2p","-CVCP3n",
"CVCP4p","-CVCP5n")
)

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

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

 #The summary scores are translated back to the original 0-10 interval
 #(R interprets the original numbers as factor levels)

 scoresCVCP<-data.frame(scaleCVCP$scores-1)
```

```
     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)
```

SA2-2 Exclusion of participants and sample selection

The task page was accessed 897 times in total. Only 659 participants entered the study, and 238 participants did not: 26 participants were flagged due to a suspicious IP-address (possible proxy-use, VPN server, or compromised server). The IP-checkers disagreed only in one of these cases. 20 participants did not submit their attention check responses, 185 participants failed at least one of the two checks in the beginning and did not enter the study, either. Two participants were not located in the United States, but in Portugal, four did not give a valid answer to the question about their state abbreviation (typical false answers were "United States of America" or "COVID-19"). One participant did not give consent and left the study.

Of those participants who started, 25 participants stopped in the general game instructions, 4 stopped their participation during the simulator instructions, and 3 stopped after the transmission game but before the end of the survey. 21 participants committed too many errors in the attention checks (at least 25) and were automatically prevented from further participation (and asked to return the task). The distribution of comprehension errors is shown in Table SA2-1.

A further 4 participants were identified to have completed part of the survey twice (e.g., they stopped their first attempt and reloaded the survey and restarted it later). Even if the second attempt might have been complete, these participants were excluded from the study. One participant was identified to need the highest amount of time to finish the survey and give inconsistent or incomprehensible responses throughout the task (e.g., espousing extremely liberal and conservative positions simultaneously). These problems were identified during the study and these places were filled by later participants. Finally, the randomization algorithm started one participant in the wrong condition at the very end of the study. We collected one additional participant in the correct condition and excluded the additional data point from the analysis (this was decided before the participant had completed the task and followed the pre-registered protocol).

SA2-3 Transmission game variables

SA2-3.1 Calculating scales

```
  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.: 232.0
##  Median : 408.0
##  Mean   : 484.9
```

Table SA2-1

Error count frequencies in comprenhension check

Errors	frequency
0	462
1	138
2	76
3	53
4	29
5	17
6	15
7	14
8	7
9	5
10	10
11	9
12	5
13	6
14	4
15	3
16	5
17	2
18	4
19	2
20	1
21	1
22	2
23	1
24	5
25	1
26	2
27	1
29	2
31	1
36	1
42	2
43	1
48	2
59	1
68	1
71	1
73	1
80	1
92	1
102	1
124	1

```
##  3rd Qu.: 648.0
##  Max.   :1000.0

  head(scoresTG)

##   Scale1
## 1    296
## 2    264
## 3    424
## 4    584
## 5    648
## 6    616
```

```
  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=600),each=25),
           condition=rep(df$conditionShortName, each=25),
           label=rep(df$PID, each=25))


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

head(roundwiseTG)
```

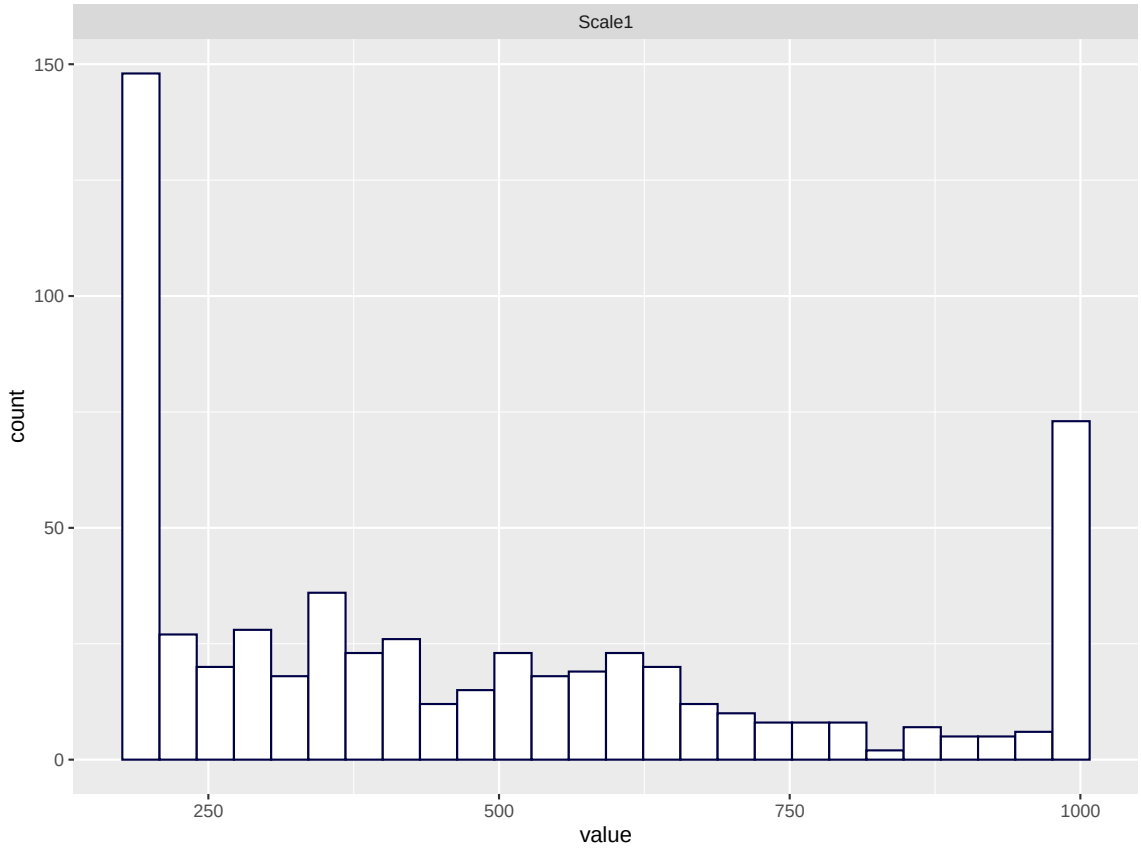

Figure SA2-1
Transmission Game results distribution

```
## # A tibble: 6 x 5
##   Round value partic condition label
##   <dbl> <int>  <int> <fct>     <fct>
## 1     1     2      1 Control   "PB001   "
## 2     2     2      1 Control   "PB001   "
## 3     3     2      1 Control   "PB001   "
## 4     4     1      1 Control   "PB001   "
## 5     5     1      1 Control   "PB001   "
## 6     6     1      1 Control   "PB001   "
```

```
plotdata= roundwiseTG %>%
    group_by(Round,condition) %>%
    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])
```

```
## `summarise()` has grouped output by 'Round'.  You can override using the
`.groups` argument.
```

```
# Percent risks to Percent safe

plotdata$counted=100-plotdata$counted
plotdata$CIlower=100-plotdata$CIlower
plotdata$CIupper=100-plotdata$CIupper
```

```
ggplot(plotdata, aes(x = Round, y = counted,color=condition)) +
  ylim(15,98) +
  ylab("Percentage of safe choices")+
  geom_line(aes(group = condition), size=3,
            alpha=0.8, position = position_dodge(width = 0.4),
            show.legend = FALSE) +
  geom_linerange(aes(ymax = CIupper, ymin = CIlower),size=0.1,alpha=0.8,
                 position = position_dodge(width = 0.4)) +
  geom_point(size = 3.5, position = position_dodge(width = 0.4),alpha=0.8)+
  scale_fill_manual(values = c("#FDE725FF","#B8DE29FF",
                    "#56C667FF","#1F968BFF","#287D8EFF","#39558CFF")) +
  scale_color_manual(values = c("#FDE725FF","#B8DE29FF",
                    "#56C667FF","#1F968BFF","#287D8EFF","#39558CFF"))+
  theme(
  panel.background = element_rect(fill = "#151515FF", colour = "#6D9EC1",
                                  size = 1, linetype = "solid"),
  panel.grid.minor = element_line(size = 0.125, linetype = 'solid',
                                  colour = "black"),
  panel.grid.major = element_line(size = 0.25, linetype = 'solid',
                                  colour = "#252525") ,
  legend.key = element_rect(colour = "transparent", fill = "white"),
  legend.background = element_rect(fill="gray",
                                   size=0.5, linetype="solid",
                                   colour ="black"),
  plot.background = element_rect(fill = "gray")

)
```

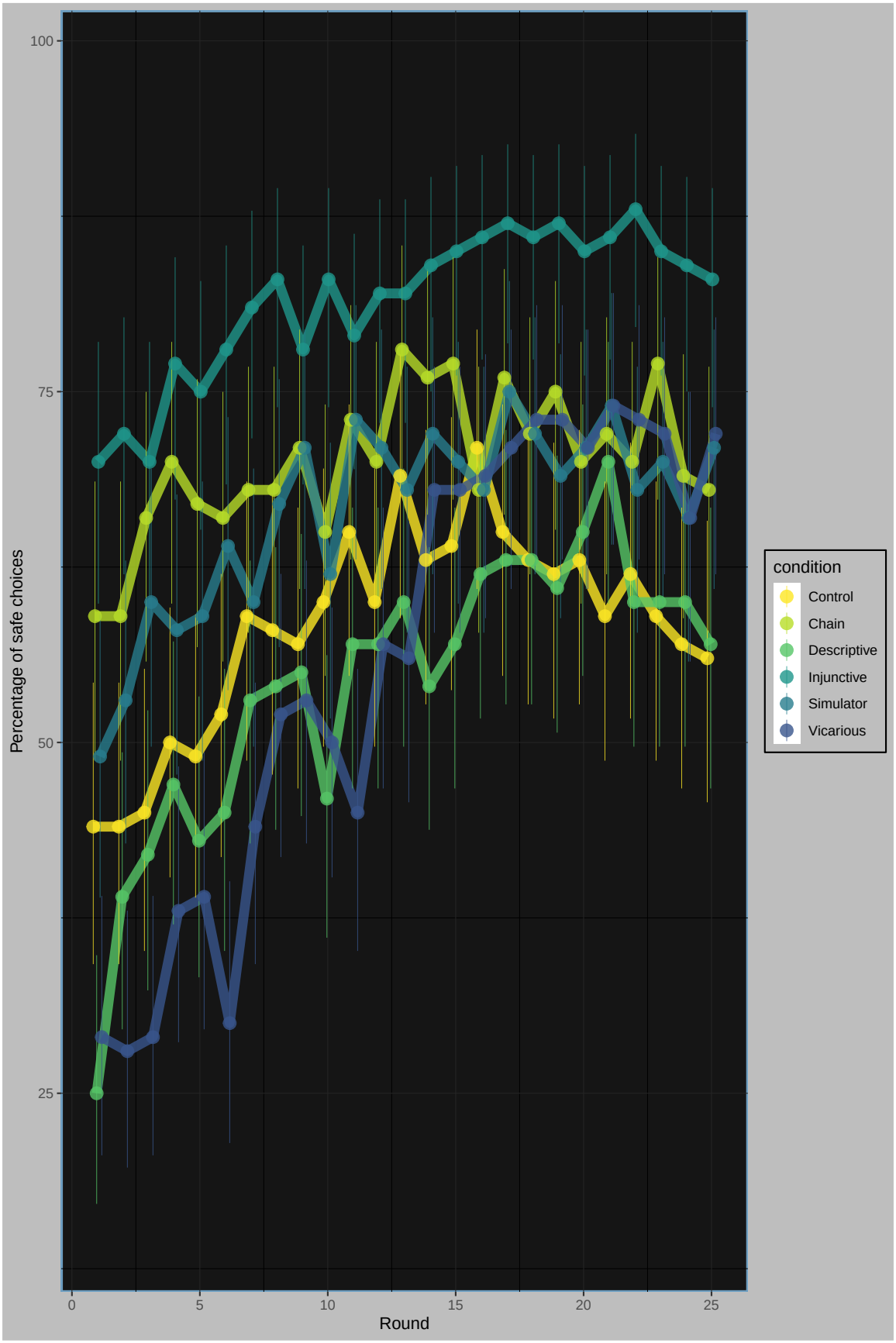

Figure SA2-2

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

The following figure uses the "ggpubr"-package and thus cannot be produced during compilation in Overleaf. Thus, we added the figure as eps-file and add the generating code as listing below.

```
library(ggpubr)
   plotdata2=data.frame(
     scoresTG,
     df$conditionShortName
   )


   plotdata2 <- plotdata2 %>%
     rename(
       condition=df.conditionShortName,
       TGscore=Scale1
     )

   plotdata2 %>%
     ggplot( aes(x = condition, y = TGscore)) +
     geom_point()


   compare_means(TGscore~condition, data= plotdata2, method = "wilcox",
       paired = FALSE,
                 ref.group = "Vicarious")


 pcomparisons=list(c("Control","Simulator"),
                   c("Control","Chain"),
                  c("Control","Injunctive"),
                  c("Control","Descriptive"),
                  c("Control","Vicarious"),
                  c("Simulator","Descriptive"),
                  c("Simulator","Vicarious")
                  )


 plotdata2 <-  plotdata2 %>%
   mutate(condition = fct_relevel(condition,
                            "Vicarious", "Descriptive", "Control",
                            "Simulator", "Chain", "Injunctive"))


 compare_means(TGscore ~ condition,  data = plotdata2)

 ggboxplot(plotdata2, x = "condition", y = "TGscore",
                  color = "condition", palette = "jco",
                  add = "jitter")+
  stat_compare_means(comparisons=pcomparisons)
```

```

corData=data.frame(
 TGscoresTG,
```

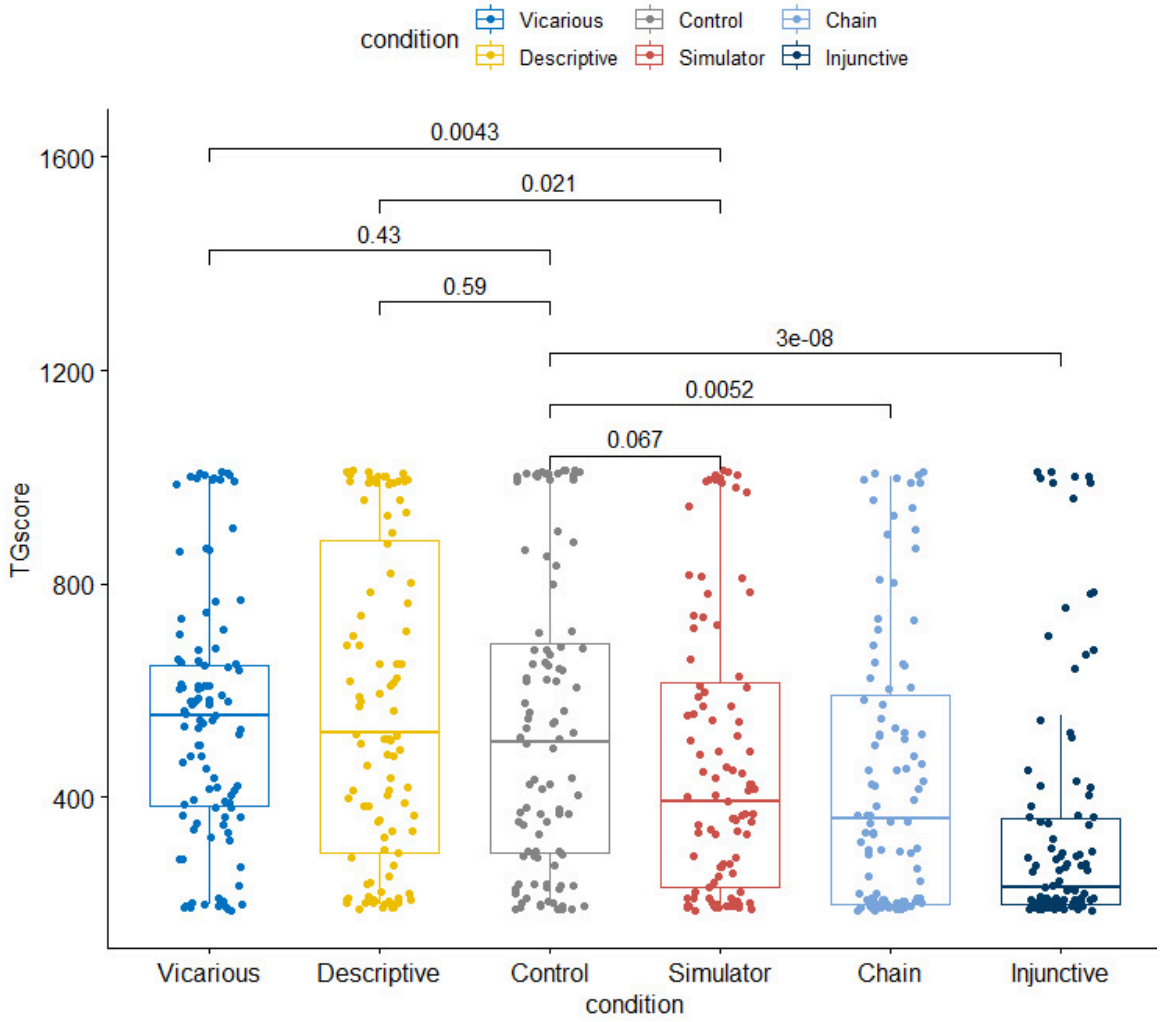

Figure SA2-3

Game results across conditions: Jittered points within the box-plots represent individual results, p-values of selected comparisons are shown.

```
  condition=df$conditionShortName
)

corData <- corData %>%
  rename(
    TGscore=Scale1
  )


 corData$PercentRisky<-((corData$TGscore-200)/32) *4
 cond=df$conditionShortName

levels(cond) <- list(CON="Control", CHN="Chain", DES="Descriptive",
                     INJ="Injunctive", SIM="Simulator",VIC="Vicarious")

levels(cond) <- list(VIC="Vicarious",  SIM="Simulator",  INJ="Injunctive",DES="Descriptive",
                     CHN="Chain",CON="Control")


levels(cond) <- list( INJ="Injunctive",
                      CHN="Chain",SIM="Simulator",CON="Control",VIC="Vicarious",DES="Descriptive")

 ggplot(corData, aes(x = PercentRisky, y = factor(cond), fill = cond)) +
  geom_density_ridges(alpha = 0.9, color = "white",
                      scale = 4, rel_min_height = 0.01,bandwidth=5) +
  labs(x = "Percentage of Risky Choices", y = "condition") +
  xlim(0,100)+
  theme_ridges()+
  theme_gray()+
  scale_fill_manual(values = c("#56C667FF","#39558CFF","#287D8EFF","#FDE725FF","#B8DE29FF",
                               "#1F968BFF"))+
  theme(legend.position = "none", axis.text.y=element_blank())+
  annotate("text", x = 80,  y = 0.85, label = "INJ", size = 6, color="black", angle=0)+
  annotate("text", x = 80,  y = 1.85, label = "CHN", size = 6, color="black", angle=0)+
  annotate("text", x = 80,  y = 2.85, label = "SIM", size = 6, color="black", angle=0)+
  annotate("text", x = 80,  y = 3.85, label = "CON", size = 6, color="black", angle=0)+
  annotate("text", x = 80,  y = 4.85, label = "VIC", size = 6, color="black", angle=0)+
  annotate("text", x = 80,  y = 5.85, label = "DES", size = 6, color="black", angle=0)
```

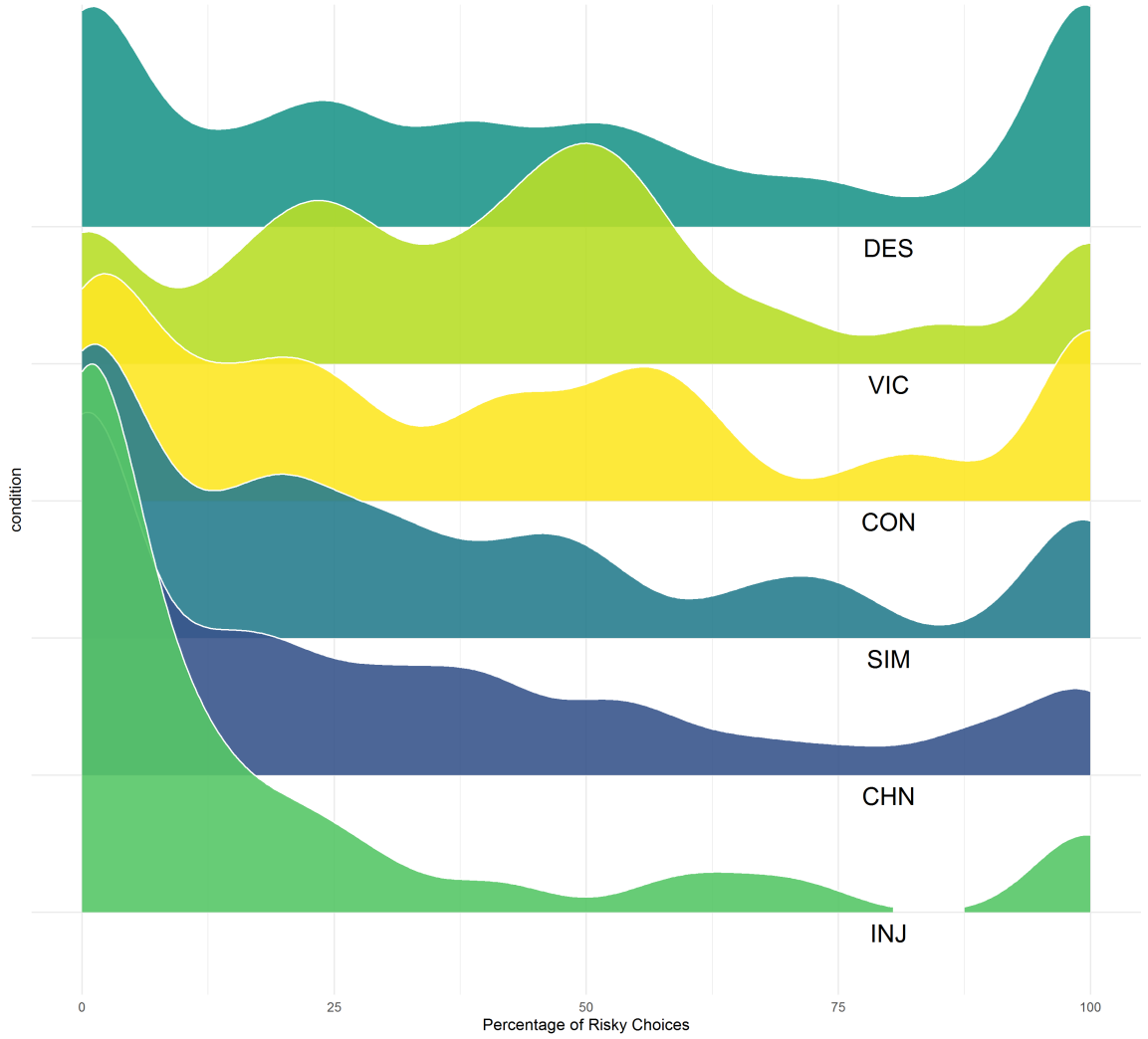

Figure SA2-4
Experimental conditions and distribution of percentage of risky choices

SA2-3.2 Classifying behavioral patterns

```
patternSwitch <- data.frame(
  S02=abs(scaleFrameTG$R02-scaleFrameTG$R01),
  S03=abs(scaleFrameTG$R03-scaleFrameTG$R02),
  S04=abs(scaleFrameTG$R04-scaleFrameTG$R03),
  S05=abs(scaleFrameTG$R05-scaleFrameTG$R04),
  S06=abs(scaleFrameTG$R06-scaleFrameTG$R05),
  S07=abs(scaleFrameTG$R07-scaleFrameTG$R06),
  S08=abs(scaleFrameTG$R08-scaleFrameTG$R07),
  S09=abs(scaleFrameTG$R09-scaleFrameTG$R08),
  S10=abs(scaleFrameTG$R10-scaleFrameTG$R09),
```

```
  S11=abs(scaleFrameTG$R11-scaleFrameTG$R10),
  S12=abs(scaleFrameTG$R12-scaleFrameTG$R11),
  S13=abs(scaleFrameTG$R13-scaleFrameTG$R12),
  S14=abs(scaleFrameTG$R14-scaleFrameTG$R13),
  S15=abs(scaleFrameTG$R15-scaleFrameTG$R14),
  S16=abs(scaleFrameTG$R16-scaleFrameTG$R15),
  S17=abs(scaleFrameTG$R17-scaleFrameTG$R16),
  S18=abs(scaleFrameTG$R18-scaleFrameTG$R17),
  S19=abs(scaleFrameTG$R19-scaleFrameTG$R18),
  S20=abs(scaleFrameTG$R20-scaleFrameTG$R19),
  S21=abs(scaleFrameTG$R21-scaleFrameTG$R20),
  S22=abs(scaleFrameTG$R22-scaleFrameTG$R21),
  S23=abs(scaleFrameTG$R23-scaleFrameTG$R22),
  S24=abs(scaleFrameTG$R24-scaleFrameTG$R23),
  S25=abs(scaleFrameTG$R25-scaleFrameTG$R24)
)

scaleSwitch=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 ),
                       items =patternSwitch,totals=TRUE)

switchNumber<-data.frame(
scaleSwitch$scores,
score=scoresTG$Scale1,
R1=scaleFrameTG$R01,
pattern=0
) %>% rename(
  switches=Scale1
)

switchNumber <- switchNumber %>%
  mutate(pattern = case_when(
    (R1=="1") & (switches ==0) ~ "p1-All G",
    (R1=="2") & (switches ==0) ~ "p4-All H",
    (R1=="2") & (switches ==1) ~ "p2-H then G",
    ((R1=="1") & (switches ==1))|(switches>1) ~"p3-Switching"
  )
  )
```

```
patterncheck=data.frame(
  CRT=scoresCRT$CRTscore,
  pattern=switchNumber$pattern,
  risk=as.numeric(df$RTGeneral),
  SVO=scoresSVO$angleSVO
)

a <-aov(CRT~pattern, data=patterncheck)
summary(a)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## pattern       3   86.3  28.781   15.45 1.09e-09 ***
## Residuals   596 1110.3   1.863
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

a <-aov(risk~pattern, data=patterncheck)
summary(a)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## pattern       3  187.7   62.55   12.89 3.62e-08 ***
## Residuals   596 2892.9    4.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

a <-aov(SVO~pattern, data=patterncheck)
summary(a)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## pattern       3   6844  2281.2   12.09 1.08e-07 ***
## Residuals   596 112455   188.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SA2-3.3 Behavioral patterns and conditions

The following command makes use of the package "ggmosaic", which is not part of Overleaf's R installation. Thus, instead of running the commands during compilation, we present the listing and include the images produced at an earlier point in time. Using the commands in R-Studio should lead to the same results (pace version differences).

```
library(ggmosaic)

patterncheck4=data.frame(
  pattern=switchNumber$pattern,
  cond=df$conditionShortName
)
```

```

patterncheck4$cond =factor(patterncheck4$cond , levels=c("Descriptive","Control",
                                        "Vicarious","Simulator","Chain","Injunctive"))

patterncheck4$pattern =factor(patterncheck4$pattern , levels=c("p4-All H",
                                            "p3-Switching","p2-H then G","p1-All G" ))


ggplot(data = patterncheck4) +
  geom_mosaic(aes(x = product(cond), fill=as.factor(pattern)), na.rm=TRUE) +
  scale_fill_viridis_d(option="D")+
  labs(x="", y='Pattern',size=6) +
  theme_dark()+
  annotate("text", x = 0.08,  y = -0.03, label = "DES", size = 9, color="black",angle=0)+
  annotate("text", x = 0.25,  y = -0.03, label = "CON", size = 9, color="white",angle=0)+
  annotate("text", x = 0.42,  y = -0.03, label = "VIC", size = 9, color="black",angle=0)+
  annotate("text", x = 0.59,  y = -0.03, label = "SIM", size = 9, color="black",angle=0)+
  annotate("text", x = 0.75,  y = -0.03, label = "CHN", size = 9, color="black",angle=0)+
  annotate("text", x = 0.92,  y = -0.03, label = "INJ", size = 9, color="black",angle=0)+
  annotate("text", x = -0.03,  y = 0.1, label = "All H", size = 7, color="black",angle=90)+
  annotate("text", x = -0.03,  y = 0.45, label = "Switching", size = 7, color="lightblue",angle=90)+
  annotate("text", x = -0.03,  y = 0.75, label = "H then G", size = 7, color="lightgreen",angle=90)+
  annotate("text", x = -0.03,  y = 0.91, label = "All G", size = 7, color="yellow",angle=90)+
   theme(legend.position = "none")
```

```
patterncheck4=data.frame(
  pattern=switchNumber$pattern,
  cond=df$conditionShortName
)
patterncheck4$cond =factor(patterncheck4$cond , levels=c("Descriptive","Control",
                                        "Vicarious","Simulator","Chain","Injunctive"))

patterncheck4$pattern =factor(patterncheck4$pattern , levels=c("p4-All H",
                                            "p3-Switching","p2-H then G","p1-All G" ))

chidata=table(as.factor(patterncheck4$pattern),as.factor(patterncheck4$cond))
```

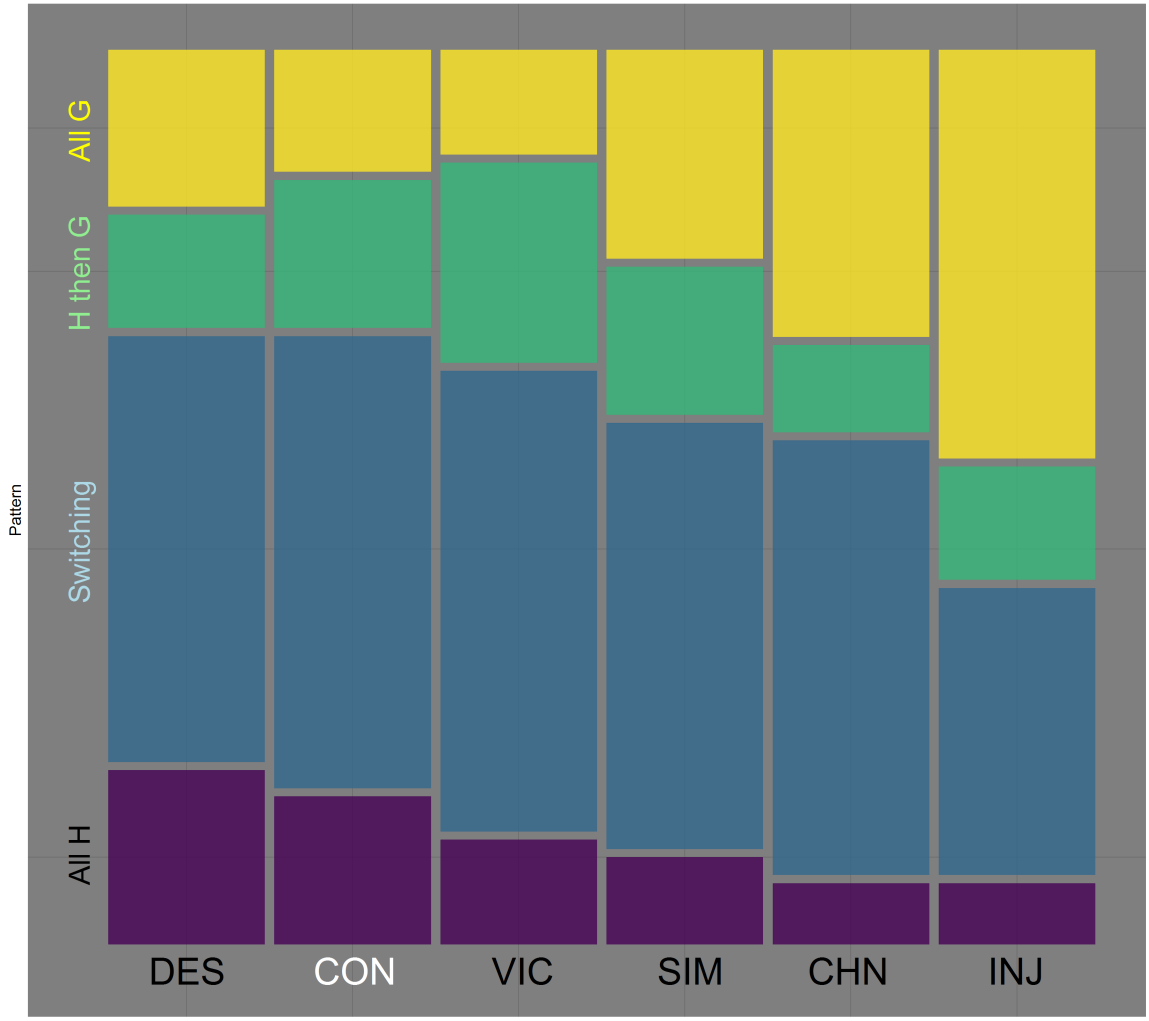

Figure SA2-5

Mosaic plot of the proportion of behavioral patterns observed in the six experimental conditions.

```
print(chidata)

##
##                Descriptive Control Vicarious Simulator Chain Injunctive
##   p4-All H              20      17        12        10     7          7
##   p3-Switching          49      52        53        49    50         33
##   p2-H then G           13      17        23        17    10         13
##   p1-All G              18      14        12        24    33         47

print(chisq.test(chidata))

##
##  Pearson's Chi-squared test
##
## data:  chidata
## X-squared = 60.082, df = 15, p-value = 2.441e-07
```

SA2-3.4 Behavioral pattern and risk preference

```
patternconditioncheck=data.frame(
  CRT=scoresCRT$CRTscore,
  pattern=switchNumber$pattern,
  risk=as.numeric(df$RTGeneral)-1,
  SVO=scoresSVO$angleSVO,
  cond=df$conditionShortName
)
a <-aov(risk~pattern, data=patterncheck)
summary(a)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## pattern       3  187.7   62.55   12.89 3.62e-08 ***
## Residuals   596 2892.9    4.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The following commands make use of the package "ggridges", which is not part of Overleaf's R installation. Thus, instead of running the commands during compilation, we present the listing and include the images produced at an earlier point in time. Using the commands in R-Studio should lead to the same results (pace version differences).

```
library(ggridges)

ggplot(patterncheck, aes(x = risk, y = factor(pattern), fill = pattern)) +
  geom_density_ridges(alpha = 0.7, color = "white",
                      scale = 5, rel_min_height = 0.02,bandwidth=1) +
  labs(x = "General risk item", y = "") +
```

```
  theme_ridges()+
  xlim(0,10)+
  theme_gray()+
  scale_y_discrete(limits = rev(levels(factor(patterncheck$pattern))))+
  scale_fill_viridis_d(option="plasma")+
  theme(legend.position = "none", axis.text.y=element_blank())+
  annotate("text", x = 0.5,  y = 0.8, label = "All H", size = 6, color="black",angle=0)+
  annotate("text", x = 0.5,  y = 1.8, label = "Switch", size = 6, color="black",angle=0)+
  annotate("text", x = 0.5,  y = 2.8, label = "H-G", size = 6, color="black", angle=0)+
  annotate("text", x = 0.5,  y = 3.8, label = "All G", size = 6, color="black",angle=0)
```

SA2-3.5 Behavioral pattern and CRT

```
resCRT <- aov(patterncheck$CRT ~patterncheck$pattern)
summary(resCRT)

##                         Df Sum Sq Mean Sq F value   Pr(>F)
## patterncheck$pattern     3   86.3  28.781   15.45 1.09e-09 ***
## Residuals              596 1110.3   1.863
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(ggmosaic)

ggplot(data = patterncheck) +
  geom_mosaic(aes(x = product(pattern), fill=as.factor(CRT)), na.rm=TRUE) +
  scale_fill_viridis_d(option="inferno")+
  labs(x="", y='CRT score',size=6) +
  theme_dark()+
  annotate("text", x = 0.12,  y = -0.03, label = "All G", size = 6, color="black",angle=0)+
  annotate("text", x = 0.325,  y = -0.03, label = "H-G", size = 6, color="black",angle=0)+
  annotate("text", x = 0.645, y = -0.03, label = "Switch", size = 6, color="black",angle=0)+
  annotate("text", x = 0.94,  y = -0.03, label = "All H", size = 6, color="black",angle=0)+
  annotate("text", x = -0.05,  y = 0.88, label = "4", size = 6, color="black", angle=0)+
  annotate("text", x = -0.05,  y = 0.64, label = "3", size = 6, color="black", angle=0)+
  annotate("text", x = -0.05,  y = 0.435, label = "2", size = 6, color="black",angle=0)+
  annotate("text", x = -0.05,  y = 0.248, label = "1", size = 6, color="black",angle=0)+
```

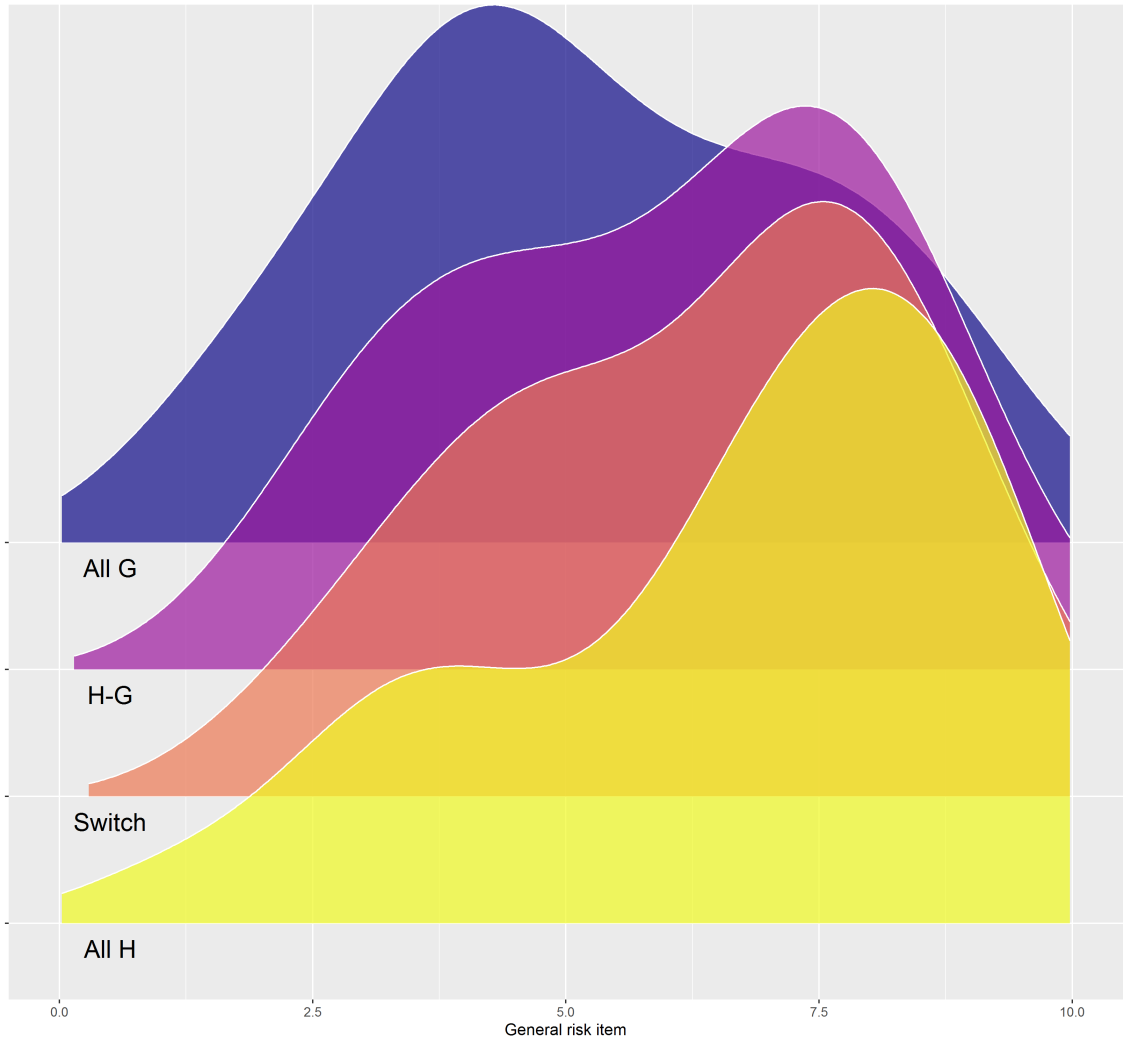

Figure SA2-6
Behavioral pattern and risk preference

```
  annotate("text", x = -0.05,  y = 0.073, label = "0", size = 6, color="black"
      ,angle=0)+
  annotate("segment", x = 0, xend=1, y = 0.5, yend=0.5,color="white")+
  annotate("text", x = 1.02,  y = 0.5, label = "Median", size = 5, color="
      white",angle=-90)+
  theme(legend.position = "none")
```

SA2-3.6 Behavioral pattern and SVO

```
a <-aov(SVO~pattern, data=patterncheck)
summary(a)
```

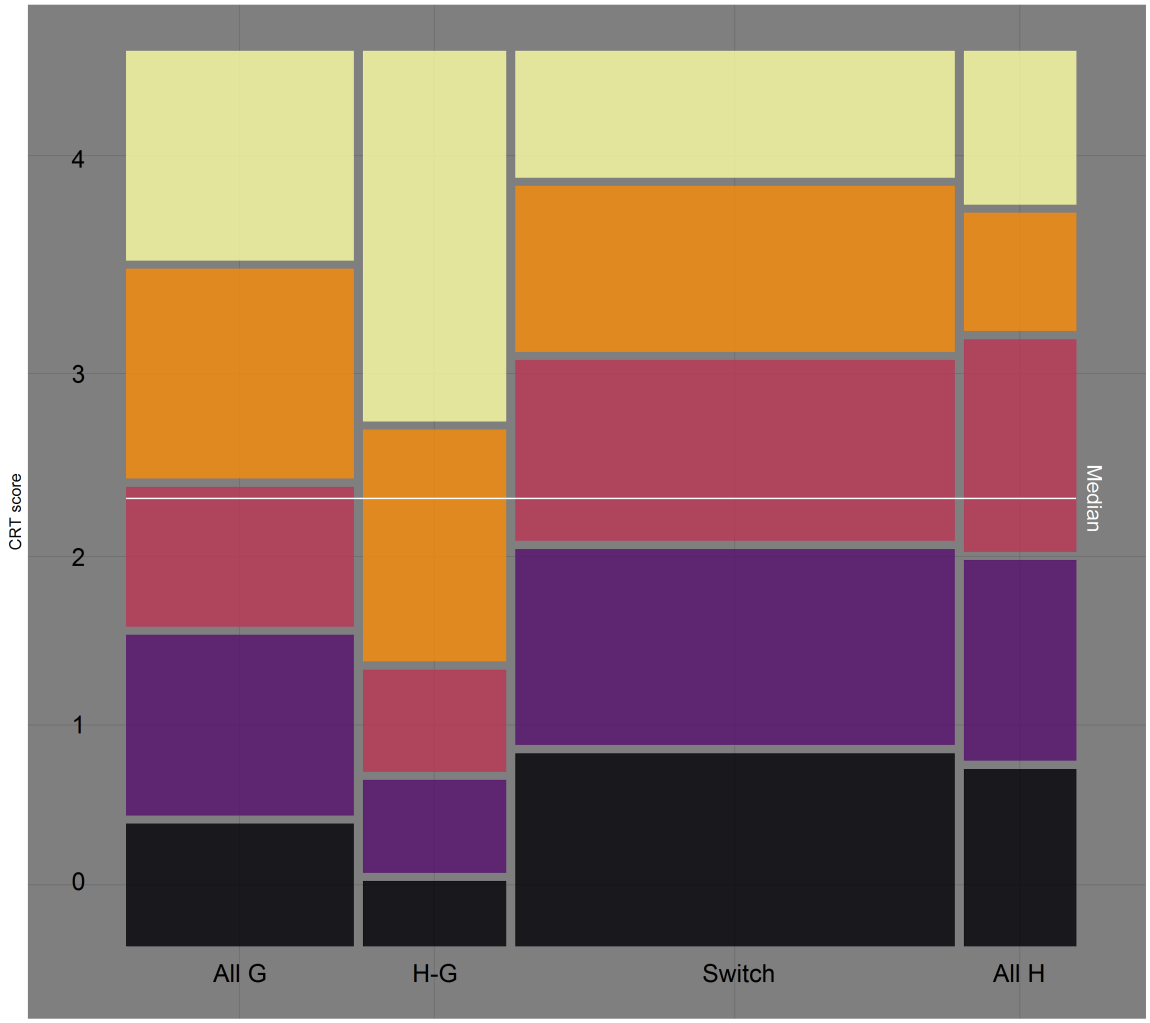

Figure SA2-7
Behavioral pattern and CRT: Mosaic plot

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## pattern       3   6844  2281.2   12.09 1.08e-07 ***
## Residuals   596 112455   188.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(patterncheck, aes(x = SVO, y = factor(pattern), fill = pattern)) +
  geom_density_ridges(alpha = 0.7, color = "white",
                      scale = 5, rel_min_height = 0.02,bandwidth=3) +
  labs(x = "SVO (angle)", y = "") +
  theme_ridges()+
  theme_gray()+
  scale_y_discrete(limits = rev(levels(factor(patterncheck$pattern))))+
  scale_fill_viridis_d(option="plasma")+
```

```
  annotate("text", x = -5,  y = 0.8, label = "All H", size = 6, color="black",
      angle=0)+
  annotate("text", x = -5,  y = 1.8, label = "Switch", size = 6, color="black"
      ,angle=0)+
  annotate("text", x = -5,  y = 2.8, label = "H-G", size = 6, color="black",
      angle=0)+
  annotate("text", x = -5,  y = 3.8, label = "All G", size = 6, color="black",
      angle=0)+
  theme(legend.position = "none", axis.text.y=element_blank())+
  xlim(-25, 55)+
  annotate(geom = "text", x = -22.5, y = 9.5, label = "Competitive",
        color = "black") +
  annotate(geom = "text", x = 5, y = 9.5, label = "Individualistic",
           color = "black")  +
  annotate(geom = "text", x = 37.5, y = 9.5, label = "Prosocial",
           color = "black") +
  annotate("segment", x = 22.45, xend = 22.45, y = 0, yend = 10,
           colour = "red")          +
  annotate("segment", x = -12.04, xend = -12.04, y = 0, yend = 10,
           colour = "red")
```

SA2-3.7 Behavioral pattern and gender

```

patterncheck3=data.frame(
  CRT=scoresCRT$CRTscore,
  pattern=switchNumber$pattern,
  risk=as.numeric(df$RTGeneral),
  SVO=scoresSVO$angleSVO,
  gender=df$demo01Gender
)

summary(patterncheck3$gender)

patterncheck3$gender <- factor(patterncheck3$gender, levels = c("Male", "
    Female", "Alternative answer:","Prefer not to say"))

ggplot(data = patterncheck3) +
  geom_mosaic(aes(x = product(pattern), fill=as.factor(gender)), na.rm=TRUE) +
  scale_fill_viridis_d(option="inferno")+
  labs(x="", y='Gender',size=6) +
  theme_dark()+
  annotate("text", x = 0.12,  y = -0.03, label = "All G", size = 6, color="
      black",angle=0)+
  annotate("text", x = 0.325,  y = -0.03, label = "H-G", size = 6, color="
      black",angle=0)+
  annotate("text", x = 0.645, y = -0.03, label = "Switch", size = 6, color="
      black",angle=0)+
  annotate("text", x = 0.94,  y = -0.03, label = "All H", size = 6, color="
      black",angle=0)+
  annotate("text", x = 0.4,  y = 0.25, label = "Male", size = 9, color="white"
      ,angle=0)+
```

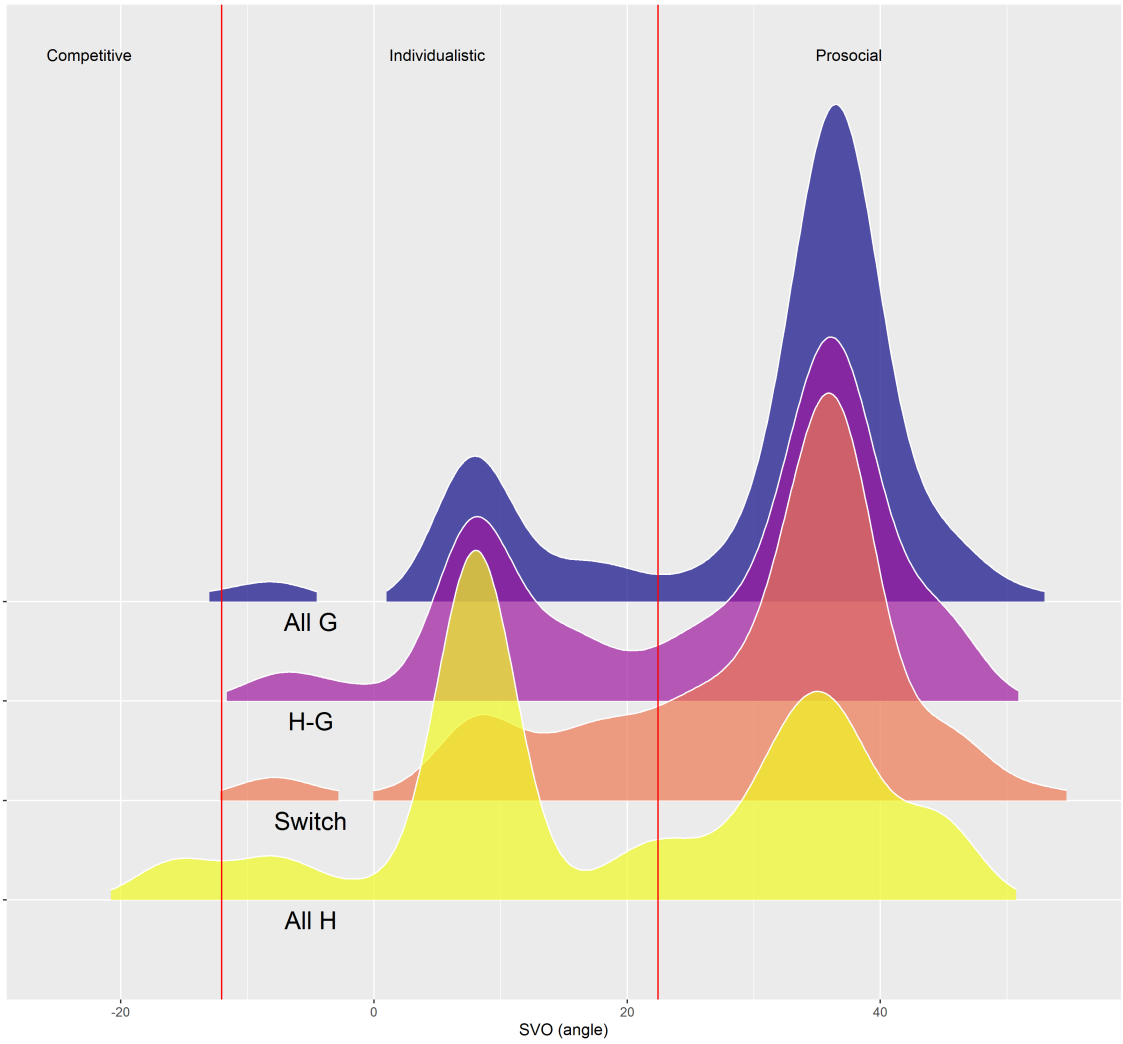

Figure SA2-8
Behavioral pattern and SVO

```
  annotate("text", x = 0.4,  y = 0.75, label = "Female", size = 9, color="
      white",angle=0)+
  annotate("text", x = 0.94,  y = 0.985, label = "Alternative", size = 3,
      color="white",angle=0)+
  annotate("text", x = 0.7,  y = 1.01, label = "Prefer not to say", size = 3,
      color="white",angle=0)+
  annotate("segment", x = 0, xend=1, y = 0.5, yend=0.5,color="blue")+
  annotate("text", x = 1.02,  y = 0.5, label = "50%", size = 5, color="blue",
      angle=-90)+
  theme(legend.position = "none")
```

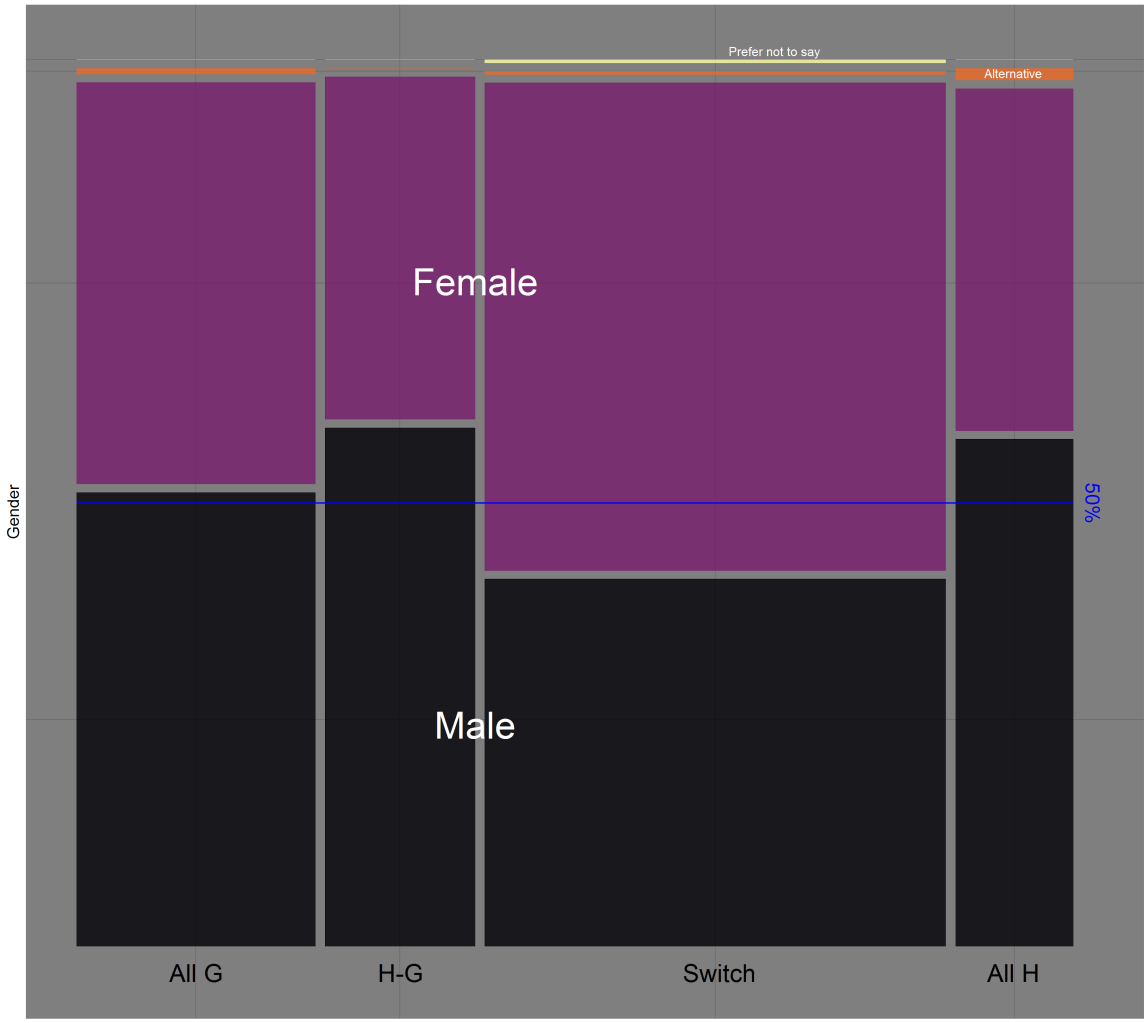

Figure SA2-9
Behavioral pattern and gender: Mosaic plot

SA2-4 Game results and predictors

SA2-4.1 Technical note

The following commands make use of the package "GGally", which is not part of Overleaf's R installation. Thus, instead of running the commands during compilation, we present the listing and include the images produced at an earlier point in time. Using the commands in R-Studio should lead to the same results (pace version differences).

For all diagrams, the following code snippet needs to have run at least once:

```
library("GGally")
cond=df$conditionShortName
levels(cond) <- list(CON="Control", CHN="Chain", DES="Descriptive",
                     INJ="Injunctive", SIM="Simulator",VIC="Vicarious")
```

All products of the "ggpairs" command (always named "p") are post-processed with the following code snippet.

```
for(i in 1:p$nrow) {
  for(j in 1:p$ncol){
     if((j>1)&(i==1)){
        p[i,j] <- p[i,j] +
            theme(panel.background = element_rect(fill = "black",
                                       colour = "darkgray",
                                       size = 2, linetype = "solid"))+
            scale_fill_manual(values = c("#FDE725FF","#B8DE29FF",
                                      "#56C667FF","#1F968BFF","#287D8EFF","
                                         #39558CFF")) +
            scale_color_manual(values = c("#FDE725FF","#B8DE29FF",
                                      "#56C667FF","#1F968BFF","#287D8EFF","
                                         #39558CFF"))
     }else if((j>1)&(i>1)&(j>i)){
        p[i,j] <- p[i,j] + theme_void() +
            scale_fill_manual(values = c("#FDE725FF","#B8DE29FF",
                                     "#56C667FF","#1F968BFF","#287D8EFF","
                                        #39558CFF")) +
            scale_color_manual(values = c("#FDE725FF","#B8DE29FF",
                                      "#56C667FF","#1F968BFF","#287D8EFF","
                                         #39558CFF"))
     }else if((j==1)| (i==j) ){
        p[i,j] <- p[i,j] + theme_dark() +

           scale_fill_manual(values = c("#FDE725FF","#B8DE29FF",
                                       "#56C667FF","#1F968BFF","#287D8EFF","
                                          #39558CFF")) +
           scale_color_manual(values = c("#FDE725FF","#B8DE29FF",
                                      "#56C667FF","#1F968BFF","#287D8EFF","
                                         #39558CFF"))
     }else{
        p[i,j] <- p[i,j] +

            scale_fill_manual(values = c("#FDE725FF","#B8DE29FF",
                                    "#56C667FF","#1F968BFF","#287D8EFF","#39558
                                       CFF")) +
           scale_color_manual(values = c("#FDE725FF","#B8DE29FF",
                                     "#56C667FF","#1F968BFF","#287D8EFF","
                                        #39558CFF"))
     }
  }
}
p
ggsave(p,file="e:/download/FILENAME.eps",width=11,height=10)
```

SA2-4.2 Predictors identified in Study 1

SA2-4.3 Exploratory analysis

```
library("GGally")

## Error in library("GGally"):  there is no package called 'GGally'
```

SA2-4.4 HEXACO-scales

```
corData=data.frame(
  scoresTG,
  scoresHEXACO$scores
)
corData <- corData %>%
  rename(
    Hl=scaleH60,
    El=scaleE60,
    Xs=scaleBX,
    As=scaleBA,
    Cs=scaleBC,
    Os=scaleBO,
    TGscore=Scale1
  )

p <- corData %>% ggpairs(
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
               discrete = "blank", combo="blank"),
  diag = list(discrete="barDiag",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()
  )
```

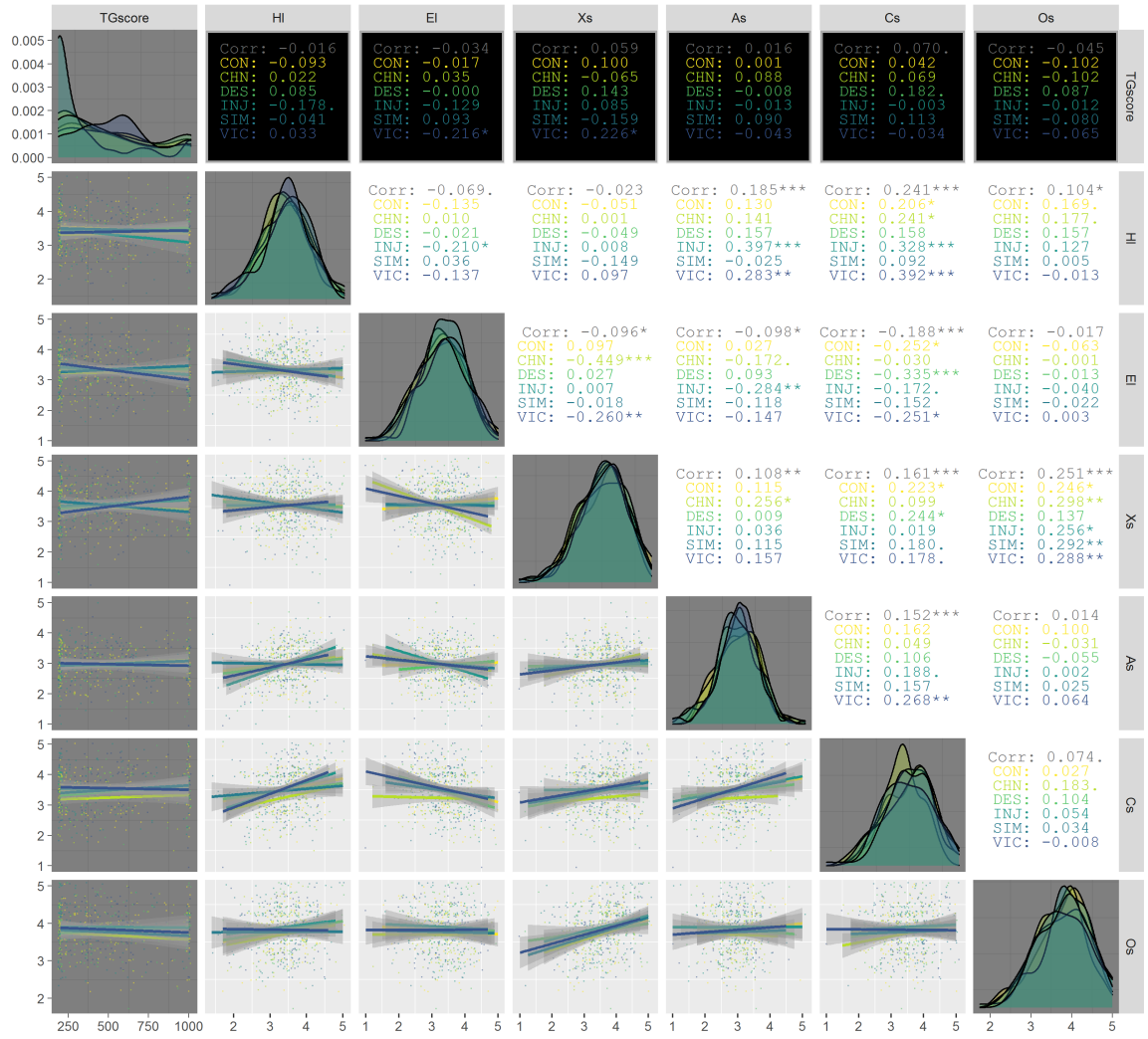

Figure SA2-10

Relationship between game results and HEXACO scales

SA2-4.5 Other personality scales

```
corData=data.frame(
  scoresTG,
  scoresPR$scores,
  Risk=as.numeric(df$RTGeneral)-1,
  Trust=as.numeric(df$TRUST)-1,
  Timepref=as.numeric(df$TIMEPREFERENCE)-1
)

corData <- corData %>%
  rename(
    scoresTG=Scale1,
    Reactance=Scale1.1
  )
```

```
14
15 p <- corData %>% ggpairs(
16   mapping = ggplot2::aes(colour=cond),
17   lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
         position_jitter()),
18                discrete = "blank", combo="blank"),
19   diag = list(discrete="barDiag",
20               continuous = wrap("densityDiag", alpha=0.3 )),
21   upper = list(combo = wrap("box_no_facet", alpha=0.5),
22                continuous = wrap("cor", size=4))) +
23   theme(panel.grid.major = element_blank()
24   )
```

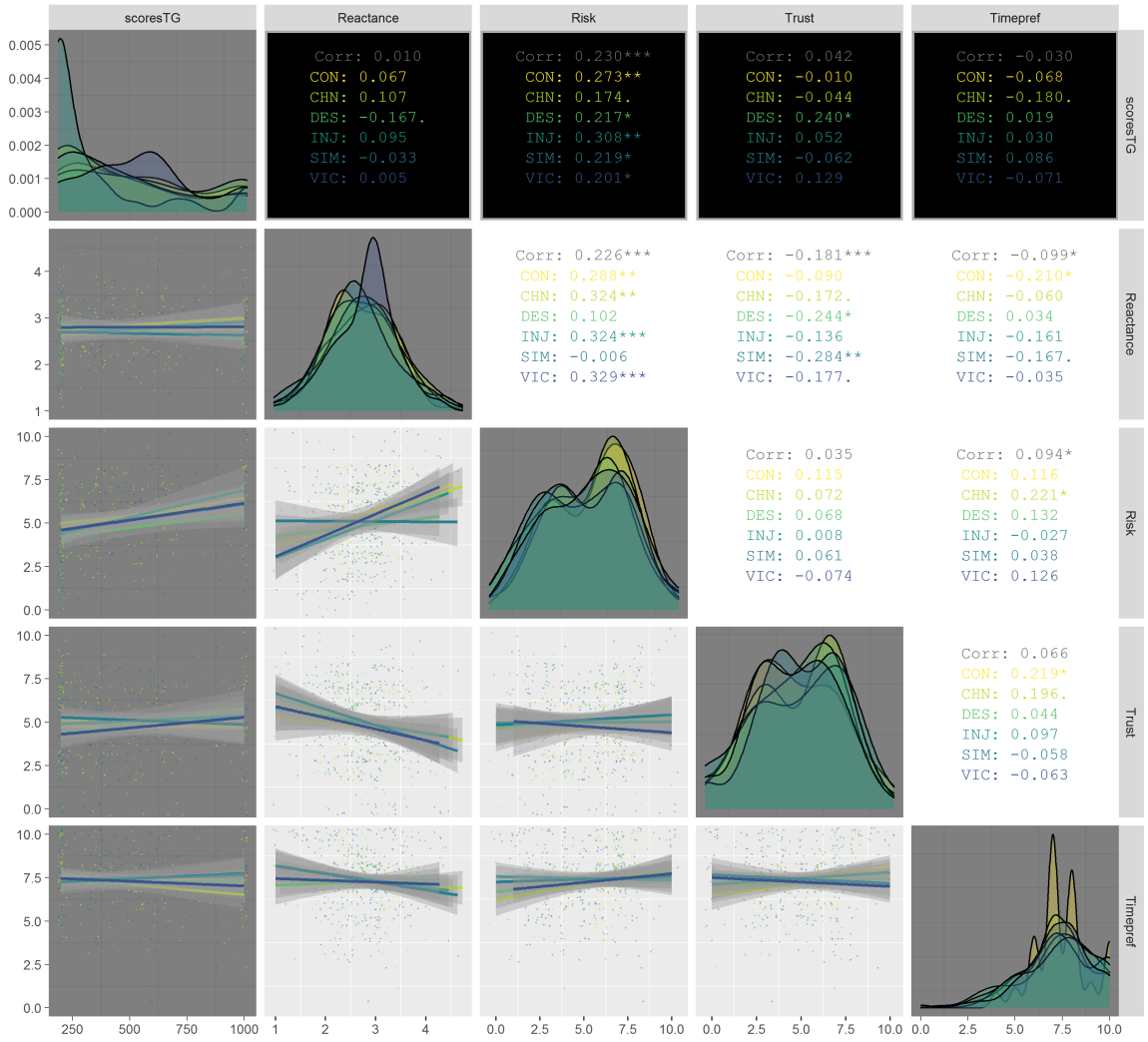

Figure SA2-11

Relationship between game results and other personality measures

SA2-4.6 Cognitive variables

```
corData=data.frame(
  scoresTG,
  scoresCRT
)

corData <- corData %>%
  rename(
    CRT=CRTscore,
    CRTi=CRTintuitive,
    TGscore=Scale1
  )

p <- corData %>% ggpairs(
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
               discrete = "blank", combo="blank"),
  diag = list(discrete="barDiag",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()
   )
```

SA2-4.7 Economic games

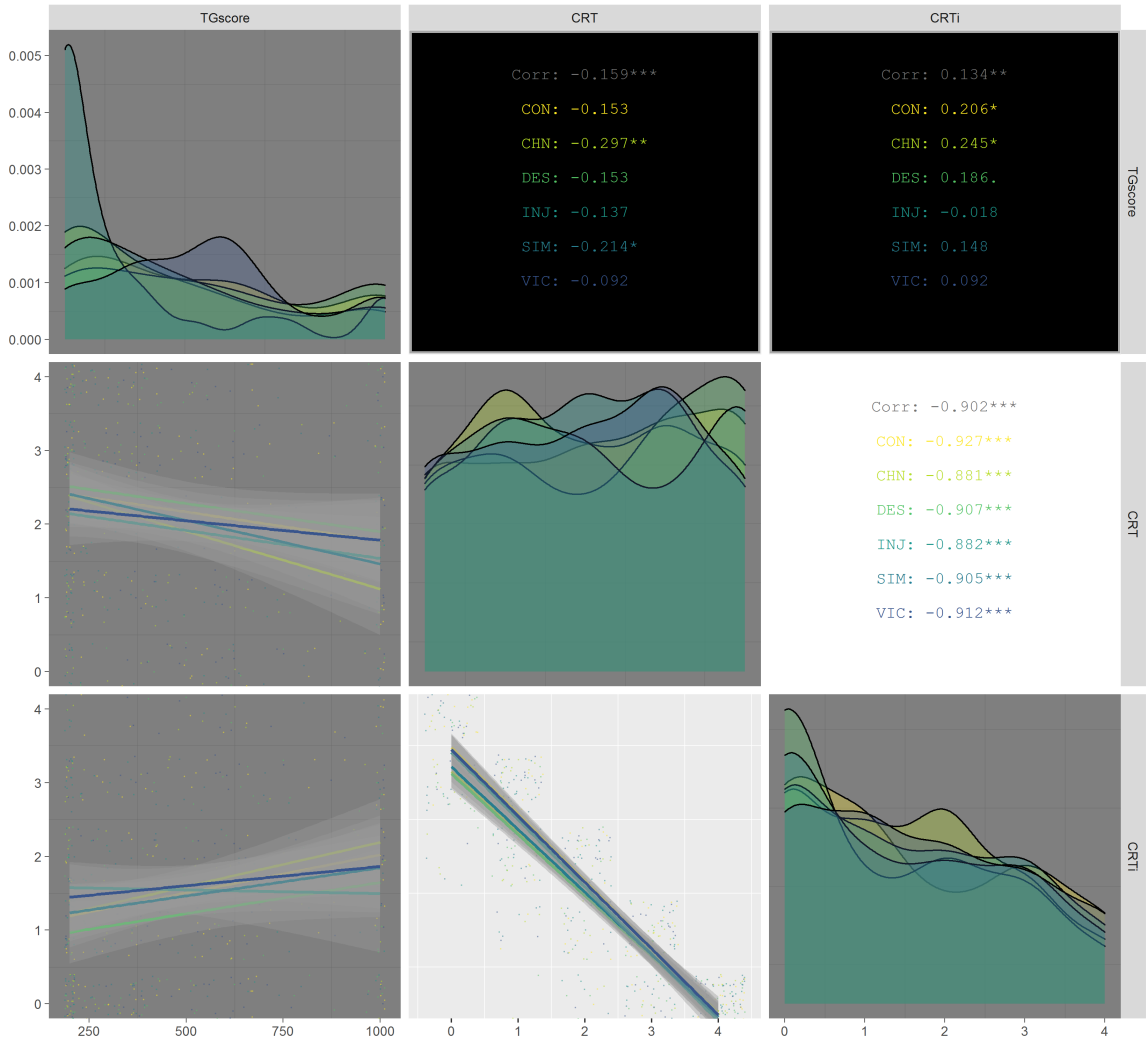

Figure SA2-12

Relationship between game results and CRT

```
1
2 corData=data.frame(
3   scoresTG,
4   angleSVO
5 )
6
7
8 corData <- corData %>%
9   rename(
10     SVO=angleSVO,
11     TGscore=Scale1
12   )
13
14
15 p <- corData %>% ggpairs(
```

```
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
                discrete = "blank", combo="blank"),
  diag = list(discrete="barDiag",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()
  )
```

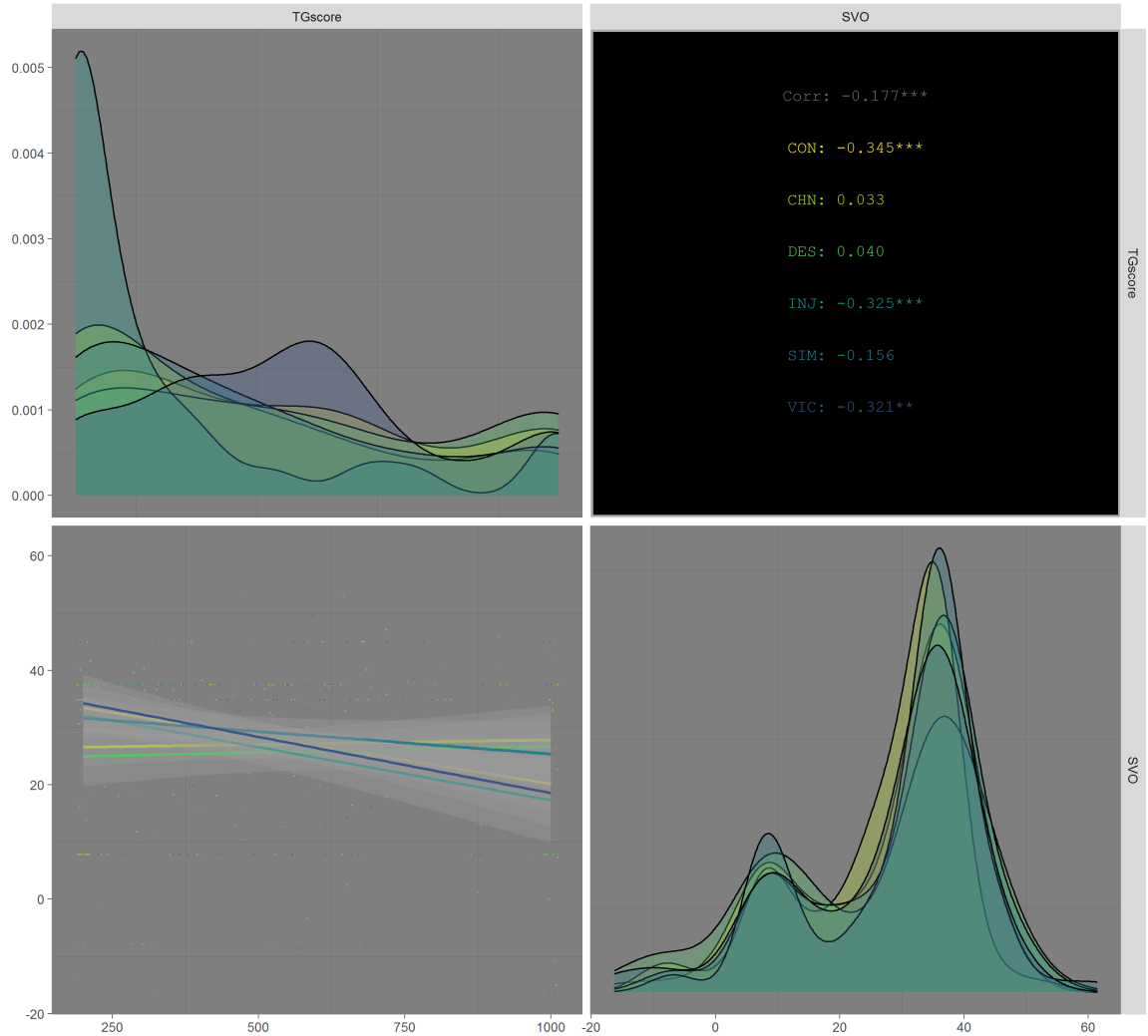

Figure SA2-13
Relationship between game results and SVO

SA2-4.8 Political measures

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


corData <- corData %>%
  rename(
    TGscore=Scale1,
    Conservatism=as.numeric.df.polPosition.,
    SECS=sclConsALL,
    SECSsoc=sclConsSoc,
    SECSecon=sclConsEcon,
    PrefTrump=Trump,
    PrefBiden=Biden

  )




p <- corData %>% ggpairs(
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
               discrete = "blank", combo="blank"),
  diag = list(discrete="barDiag",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()

  )
```

SA2-4.9 Categorical variables

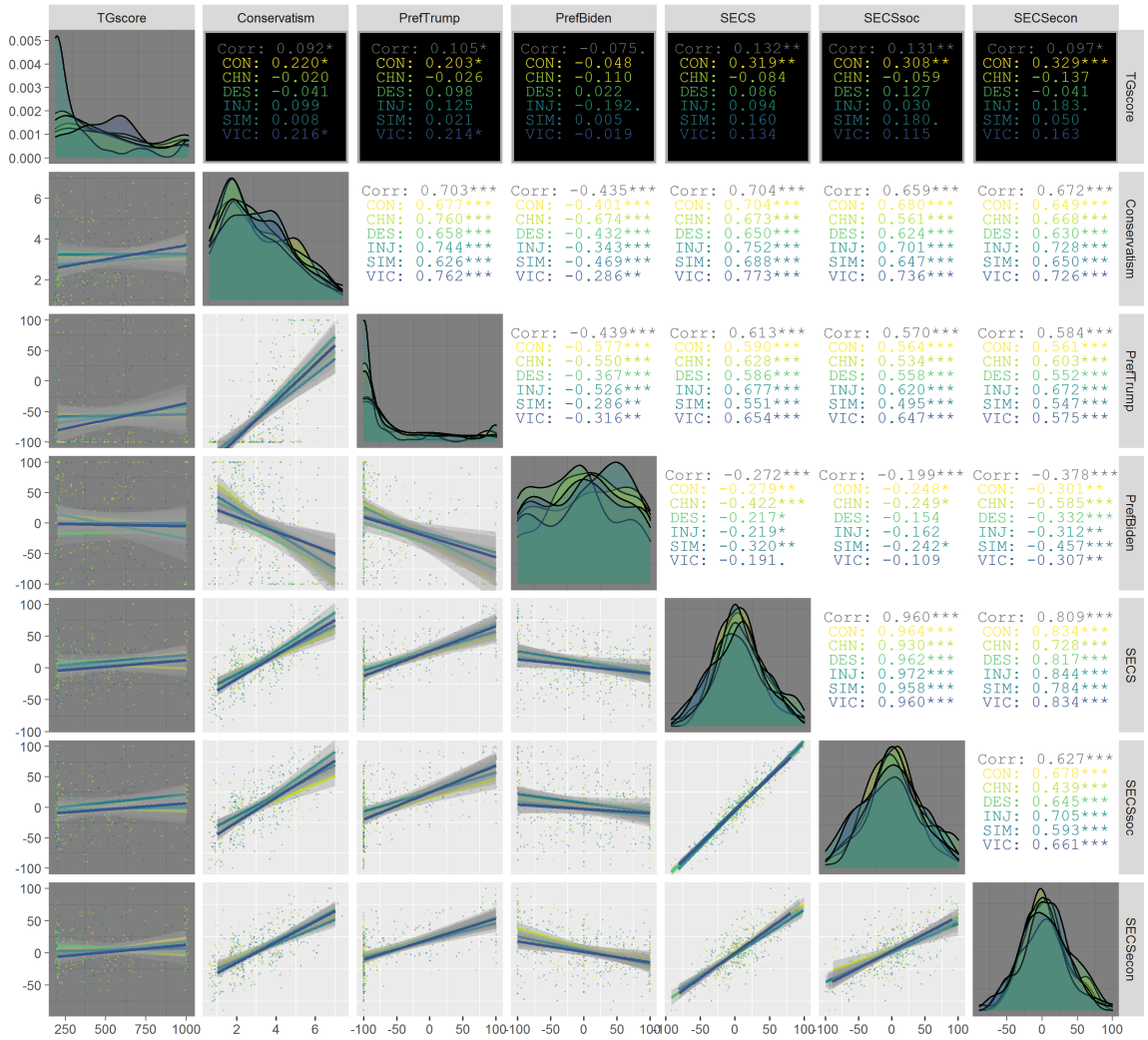

Figure SA2-14

Relationship between game results and political measures

SA2-4.9.1 Demographics

```
library(dplyr)

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

group_by(anovaFrame, gender) %>%
```

```
  filter(as.numeric(gender) > 1) %>%
summarise(
  count = n(),
  mean = mean(Scale1, na.rm = TRUE),
  sd = sd(Scale1, na.rm = TRUE)
)
```

```
## # A tibble: 3 x 4
##   gender              count  mean    sd
##   <fct>               <int> <dbl> <dbl>
## 1 Male                  299  486.  281.
## 2 Female                297  482.  268.
## 3 Alternative answer:     3  637.  405.
```

```
res.aov <- aov(Scale1 ~ gender, data=anovaFrame %>%
                 filter(as.numeric(gender) > 1))
summary(res.aov)
```

```
##              Df   Sum Sq Mean Sq F value Pr(>F)
## gender        2    73518   36759   0.485  0.616
## Residuals   596 45136465   75732
```

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

Gender.

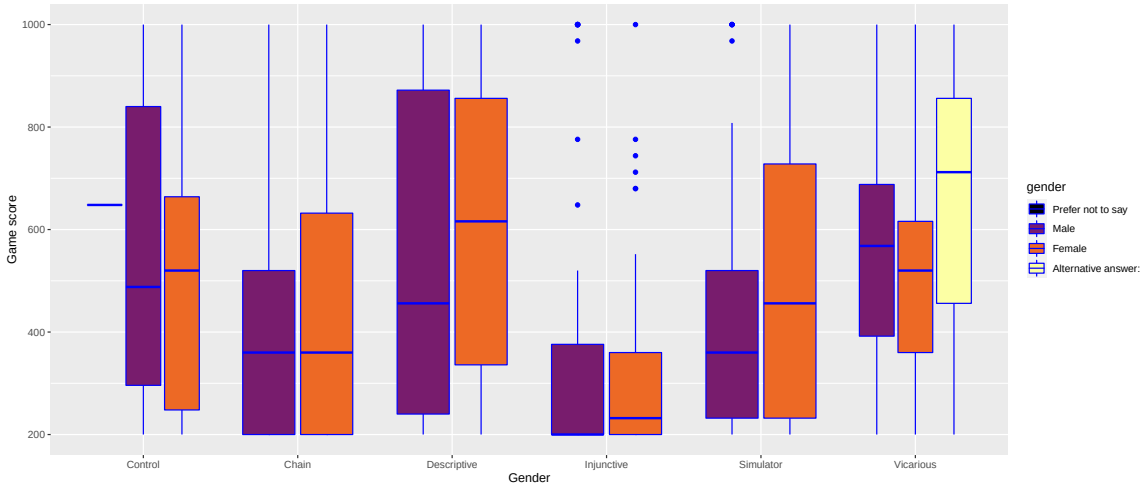

Figure SA2-15
Game scores by gender and condition

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

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

## # A tibble: 6 x 4
##   education                   count  mean    sd
##   <fct>                       <int> <dbl> <dbl>
## 1 Some High School                4  360   238.
## 2 High School                   193  494.  278.
## 3 Bachelor's degree             270  478.  273.
## 4 Master's degree                96  515   282.
## 5 PhD / MD / doctorate degree    20  432   247.
## 6 Professional degree            17  413.  272.

res.aov <- aov(Scale1 ~ education, data=anovaFrame %>%
                 filter(as.numeric(education) > 1))
summary(res.aov)
```

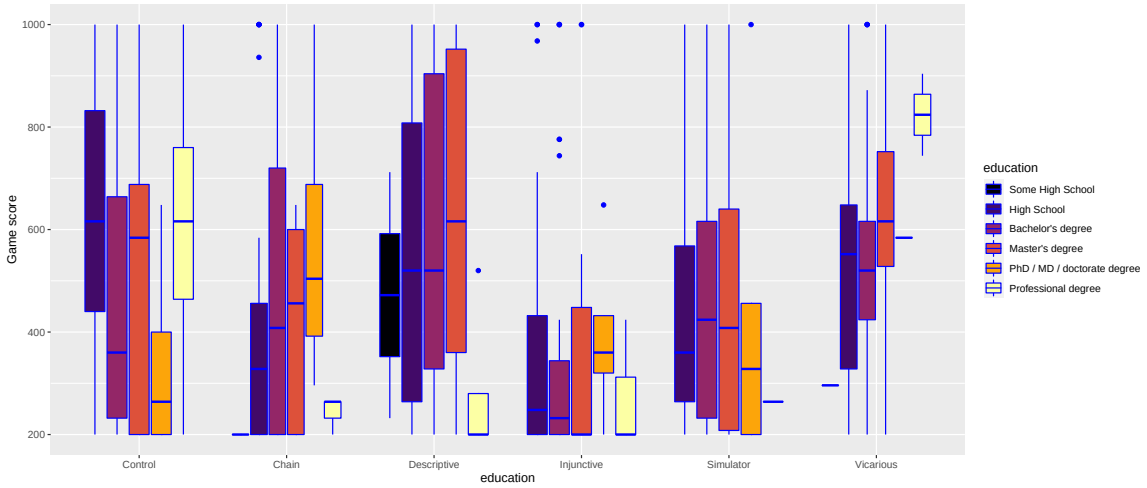

Figure SA2-16
Game scores by education and condition

```
##               Df    Sum Sq Mean Sq F value Pr(>F)
## education      4    257684   64421   0.851  0.493
## Residuals    591 44746133   75713
```

```
ggplot(anovaFrame, aes(x = condition, y = Scale1, fill=education  )) +
  geom_boxplot(color="blue")+
  labs(x="education",y="Game score") +
  scale_fill_viridis_d(option="inferno")
```

Education.

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

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

## # A tibble: 5 x 4
##   income              count  mean    sd
```

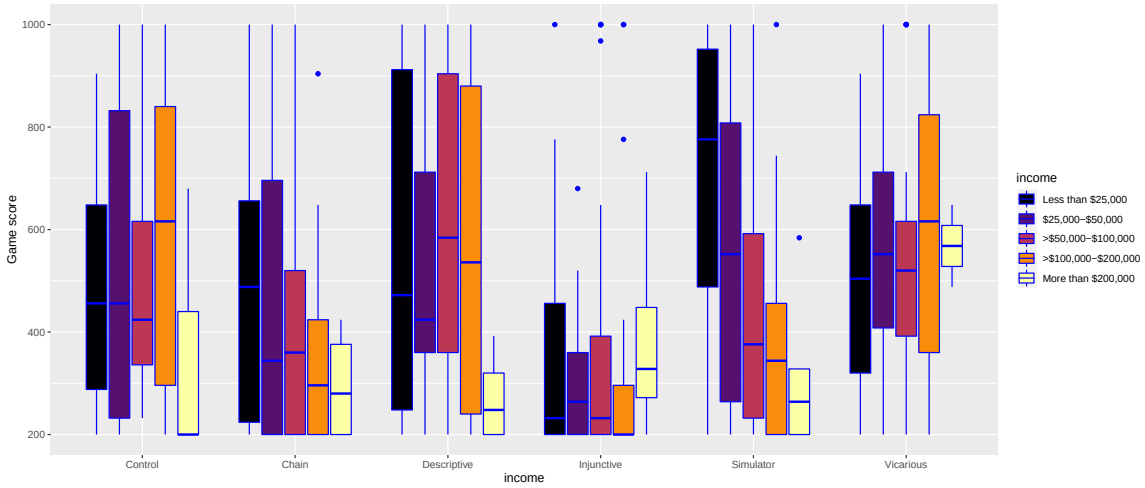

Figure SA2-17
Game scores by HH income and condition

```
##   <fct>              <int> <dbl> <dbl>
## 1 Less than $25,000     92  490.  276.
## 2 $25,000-$50,000      144  509.  281.
## 3 >$50,000-$100,000    203  489.  270.
## 4 >$100,000-$200,000   138  474.  284.
## 5 More than $200,000    23  343.  171.
```

```
res.aov <- aov(Scale1 ~ income, data=anovaFrame %>%
                 filter(as.numeric(income) > 1))
summary(res.aov)
```

```
##              Df   Sum Sq Mean Sq F value Pr(>F)
## income        3   560806  186935   2.497 0.0591 .
## Residuals   504 37736729   74874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(anovaFrame, aes(x = condition, y = Scale1, fill=income  )) +
  geom_boxplot(color="blue")+
  labs(x="income",y="Game score") +
  scale_fill_viridis_d(option="inferno")
```

Household income (2019).

SA2-4.9.2 Religion and politics

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

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

## # A tibble: 2 x 4
##   religion count  mean    sd
##   <fct>    <int> <dbl> <dbl>
## 1 no         399  482.  276.
## 2 yes        201  491.  274.

res.aov <- aov(Scale1 ~ religion, data=anovaFrame %>%
                 filter(as.numeric(religion) > 1))

## Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]):
contrasts can be applied only to factors with 2 or more levels

summary(res.aov)

##              Df   Sum Sq Mean Sq F value Pr(>F)
## income        3   560806  186935   2.497 0.0591 .
## Residuals   504 37736729   74874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ggplot(anovaFrame, aes(x = condition, y = Scale1, fill=religion  )) +
  geom_boxplot(color="blue")+
  labs(x="religion",y="Game score") +
  scale_fill_viridis_d(option="inferno")
```

Religion.

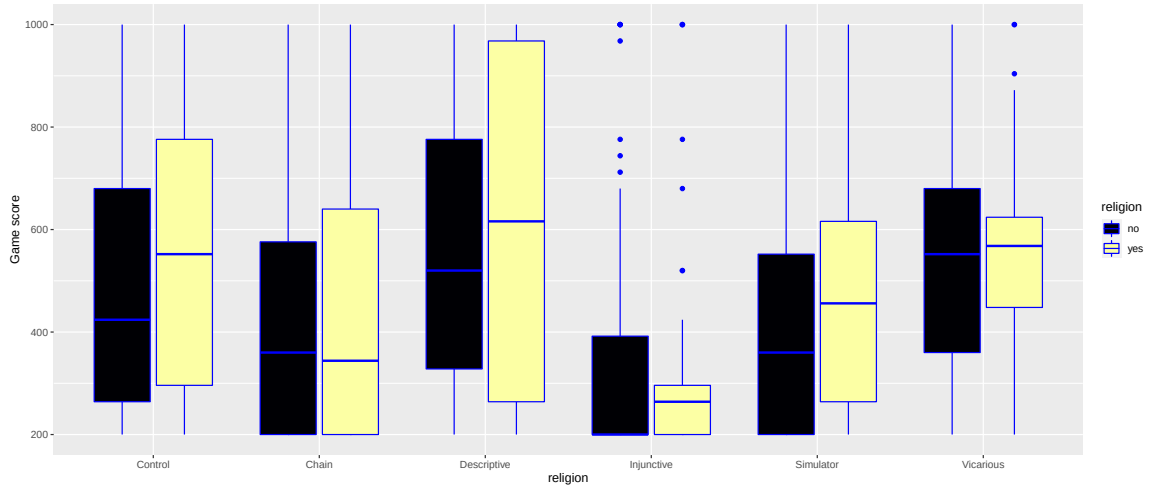

Figure SA2-18
Game scores by religiosity and condition

```
anovaFrame=data.frame(
      PolParty=df$polParty1,
  condition=df$conditionShortName,
  scoresTG
)

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


## # A tibble: 4 x 4
##   PolParty    count  mean    sd
##   <fct>       <int> <dbl> <dbl>
## 1 Other          27  529.  298.
## 2 Republican     81  505.  294.
## 3 Democrat      312  481.  271.
## 4 Independent   180  475.  271.


res.aov <- aov(Scale1 ~ PolParty, data=anovaFrame %>%
                 filter(as.numeric(PolParty) > 1))
summary(res.aov)
```

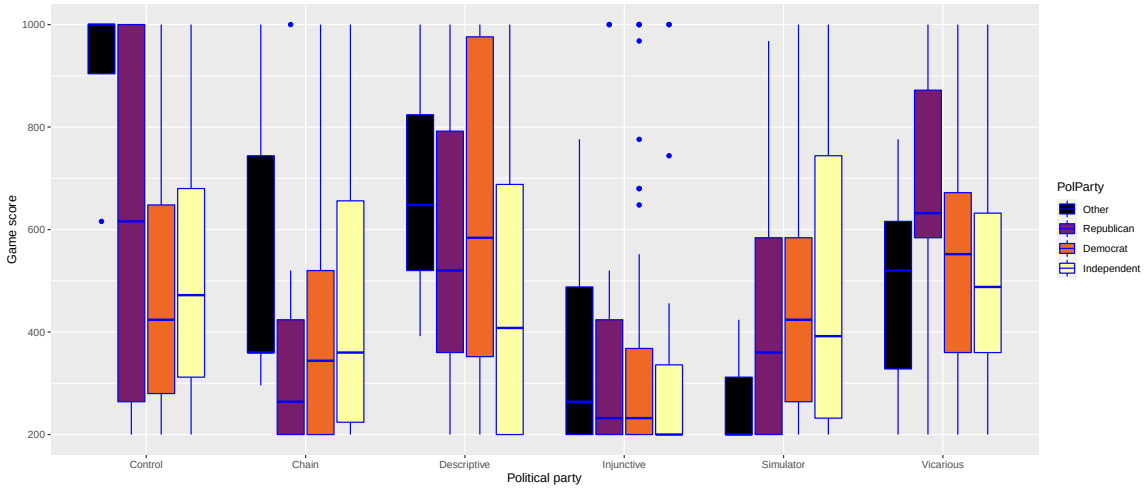

Figure SA2-19
Game scores by political party affiliation and condition

```
##                Df    Sum Sq Mean Sq F value Pr(>F)
## PolParty        2     52778   26389   0.351  0.704
## Residuals     570 42815862   75116
```

```
ggplot(anovaFrame, aes(x = condition, y = Scale1, fill=PolParty  )) +
  geom_boxplot(color="blue")+
  labs(x="Political party",y="Game score") +
  scale_fill_viridis_d(option="inferno")
```

Political party.

SA2-5 Game behavior and postquestionnaire

SA2-5.1 Motivation

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

- **Post01**: I wanted to make as much bonus money as possible.
- **Post02**: I wanted to make more bonus money than other players.
- **Post03**: I felt responsible for other players.
- **Post04**: I wanted to make other players switch color.
- **Post05**: I was afraid to switch color in this game.
- **Post06**: I tried to anticipate what others were doing in this game.
- **Post07**: I followed my gut in this game.

- **Post08**: I wanted to take some risk in this game.
- **Post09**: I wanted to protect others in this game.[1]
- **Post10**: I did not care at all what happened in this game.
- **Post11**: I wanted that the entire group receives as much money as possible.

The items were split in two parts and correlated with the game score below—again visualized using the GGally-library.

```
1
2  corData=data.frame(
3    scoresTG,
4    as.numeric(df$PostTG_01),
5    as.numeric(df$PostTG_02),
6    as.numeric(df$PostTG_03),
7    as.numeric(df$PostTG_04),
8    as.numeric(df$PostTG_05),
9    as.numeric(df$PostTG_06)
10 )
11
12 corData <- corData %>%
13   rename(
14     MaxBonus=as.numeric.df.PostTG_01.,
15     MoreBonus=as.numeric.df.PostTG_02.,
16     Responsible=as.numeric.df.PostTG_03.,
17     MakeSwitch=as.numeric.df.PostTG_04.,
18     AfraidSwitch=as.numeric.df.PostTG_05.,
19     Anticipate=as.numeric.df.PostTG_06.,
20     TGscore=Scale1
21   )
22
23 p <- corData %>% ggpairs(
24   mapping = ggplot2::aes(colour=cond),
25   lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
         position_jitter())),
26                discrete = "blank", combo="dot"),
27   diag = list(discrete="barDiag", combo="denstrip",
28               continuous = wrap("densityDiag", alpha=0.3 )),
29   upper = list(combo = wrap("box_no_facet", alpha=0.5),
30                continuous = wrap("cor", size=4))) +
31   theme(panel.grid.major = element_blank()
32
33   )
```

[1]The item is named "PostTG1_09" in the dataset, which should have been "PostTG_09".

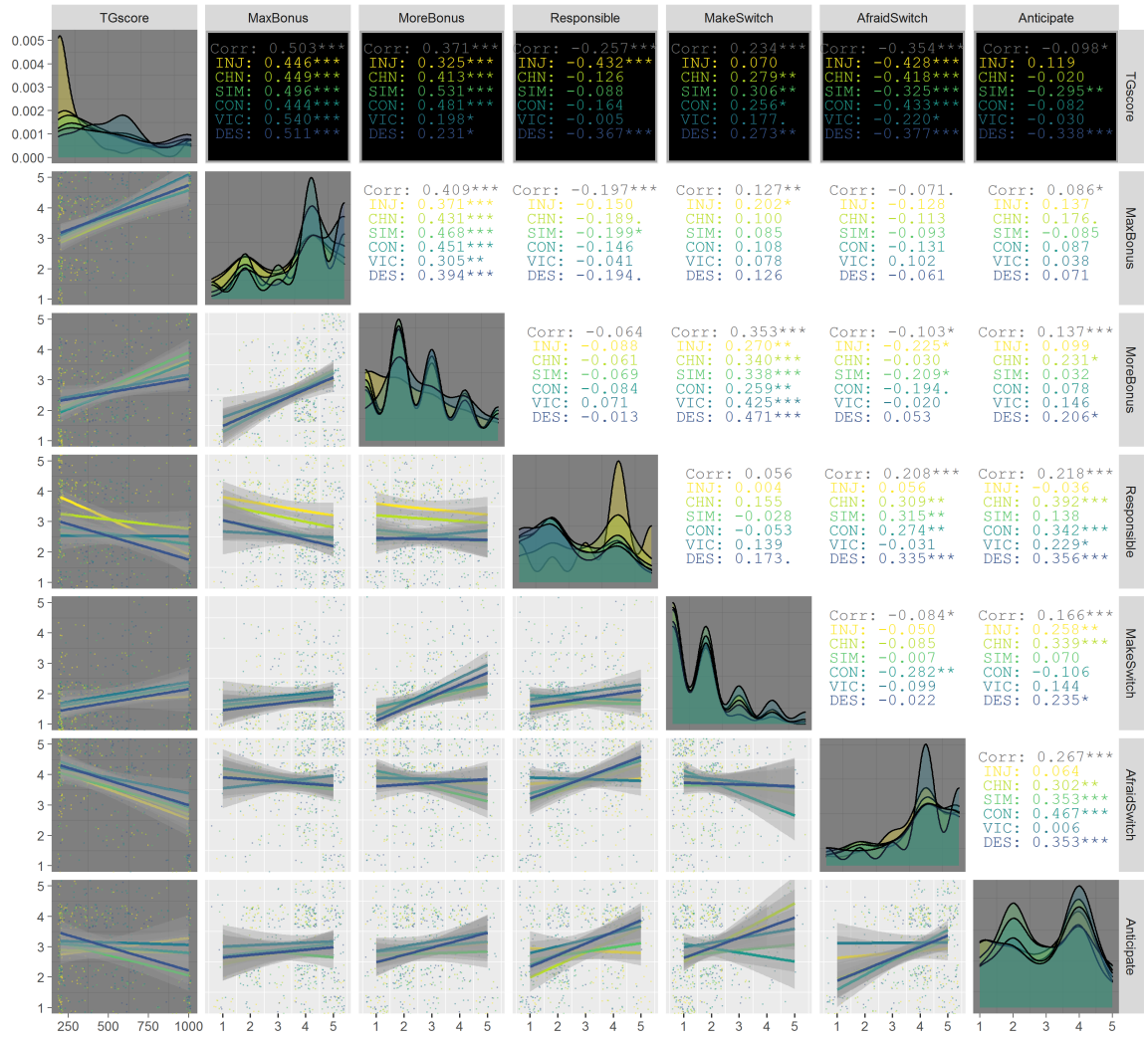

Figure SA2-20

Relationship between game results and postquestionnaire motivations (part 1)

```
1
2  corData=data.frame(
3    scoresTG,
4    as.numeric(df$PostTG_07),
5    as.numeric(df$PostTG_08),
6    as.numeric(df$PostTG1_09),
7    as.numeric(df$PostTG_10),
8    as.numeric(df$PostTG_11)
9  )
10
11 corData <- corData %>%
12   rename(
13     FollowGut=as.numeric.df.PostTG_07.,
14     TakeRisk=as.numeric.df.PostTG_08.,
15     ProtectOthers=as.numeric.df.PostTG1_09.,
```

```
    DidNotCare=as.numeric.df.PostTG_10.,
    AllReceive=as.numeric.df.PostTG_11.,
    TGscore=Scale1
  )


p <- corData %>% ggpairs(
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
               discrete = "blank", combo="dot"),
  diag = list(discrete="barDiag", combo="denstrip",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()

  )
```

SA2-5.2 Motivation and risk preference

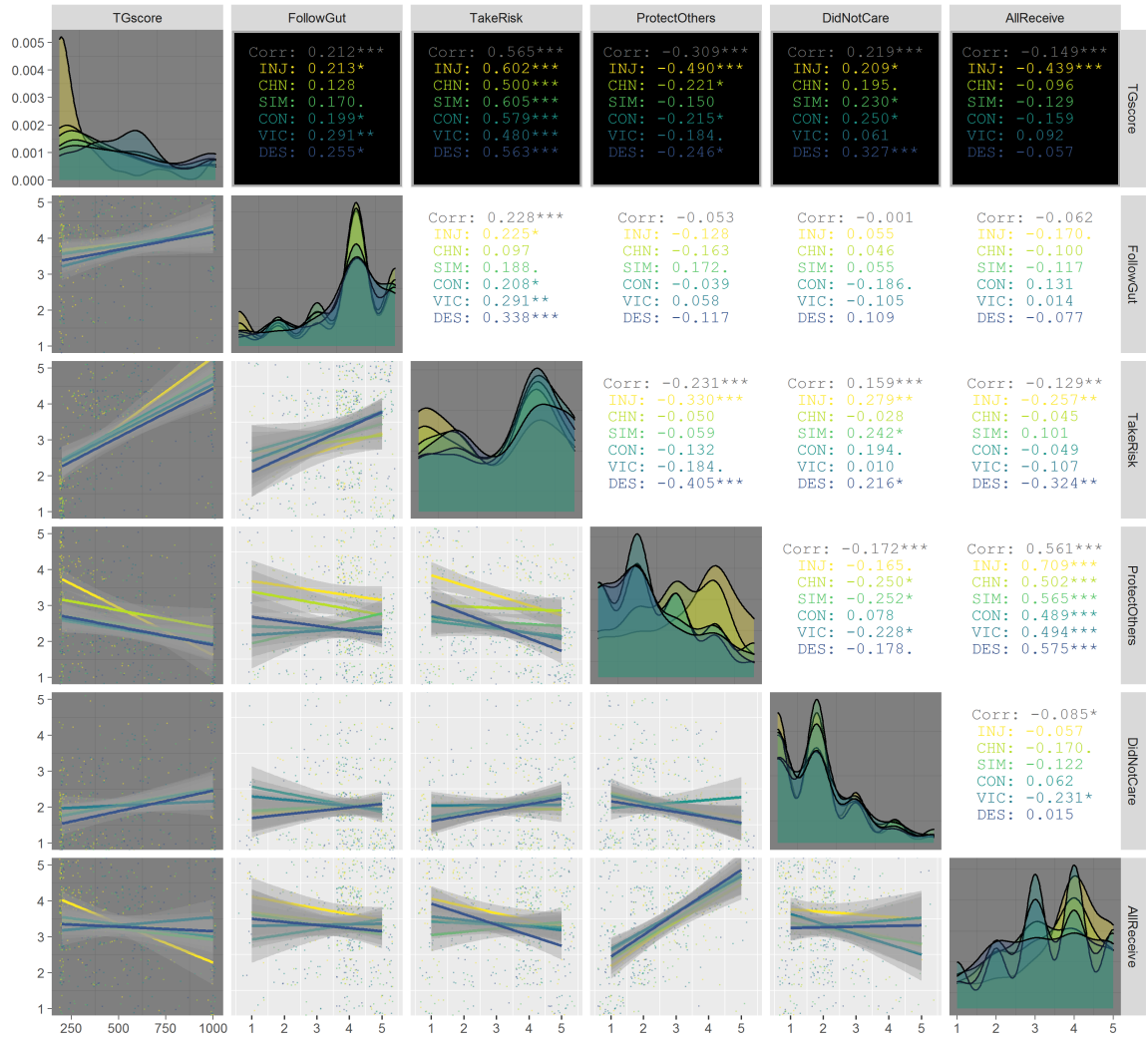

Figure SA2-21

Relationship between game results and postquestionnaire motivations (part 2)

```
corData=data.frame(
  as.numeric(df$RTGeneral),
  as.numeric(df$PostTG_05),
  as.numeric(df$PostTG_08),
  scoresTG
)

corData <- corData %>%
  rename(
    AfraidSwitch=as.numeric.df.PostTG_05.,
    TakeRisk=as.numeric.df.PostTG_08.,
    Risk=as.numeric.df.RTGeneral.,
    TGscore=Scale1
  )

```

```
16
17 p <- corData %>% ggpairs(
18   mapping = ggplot2::aes(colour=cond),
19   lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
         position_jitter()),
20                discrete = "blank", combo="dot"),
21   diag = list(discrete="barDiag", combo="denstrip",
22               continuous = wrap("densityDiag", alpha=0.3 )),
23   upper = list(combo = wrap("box_no_facet", alpha=0.5),
24                continuous = wrap("cor", size=4))) +
25   theme(panel.grid.major = element_blank()
26
27   )
```

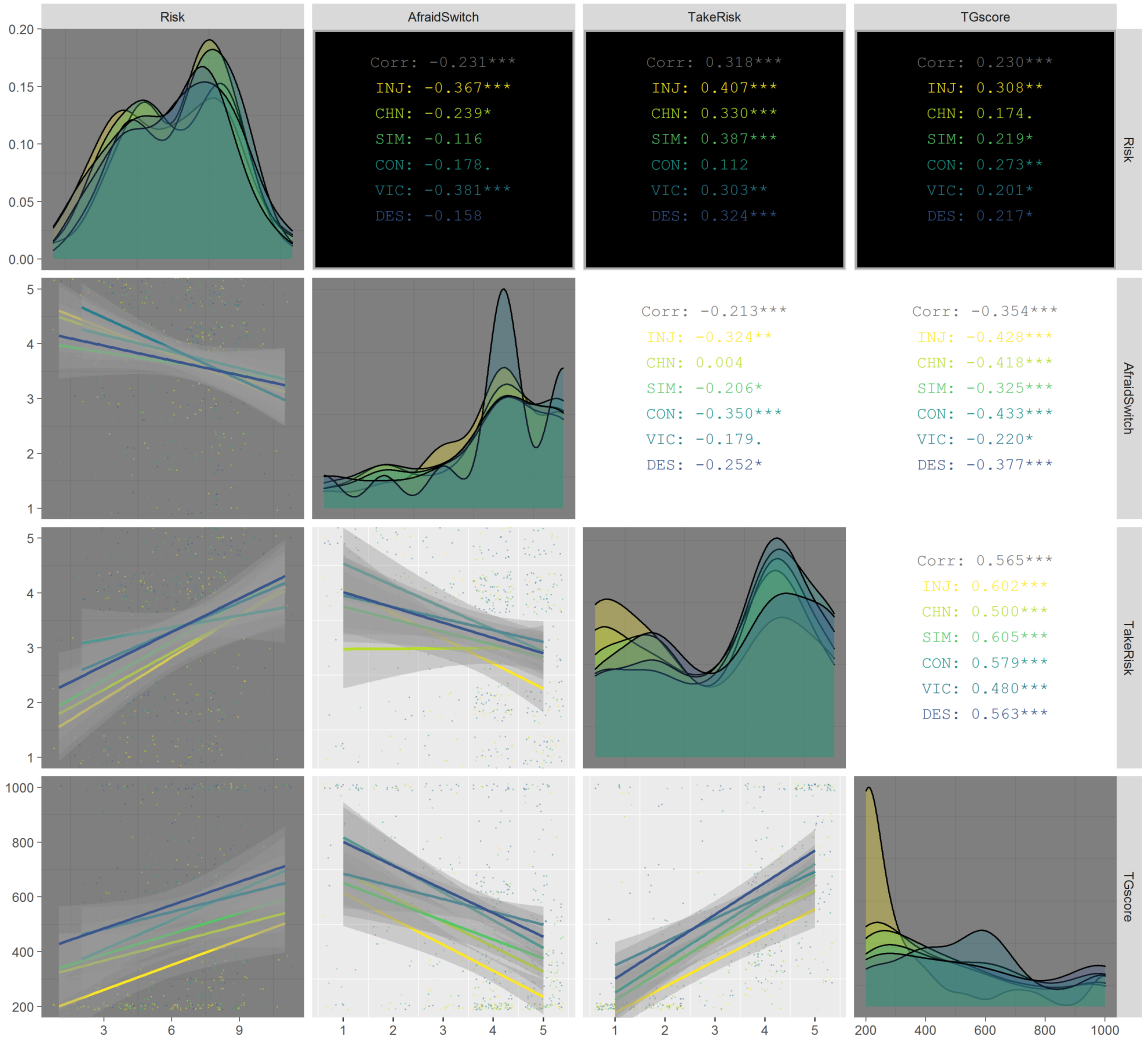

Figure SA2-22

Motivation and risk preference

SA2-5.3 Motivation and SVO

```
1
2  corData=data.frame(
3    SVO=scoresSVO$angleSVO,
4    as.numeric(df$PostTG_01),
5    as.numeric(df$PostTG_02),
6    as.numeric(df$PostTG_03),
7    as.numeric(df$PostTG1_09),
8    as.numeric(df$PostTG_11),
9    scoresTG
10 )
11
12 corData <- corData %>%
13   rename(
14     MaxBonus=as.numeric.df.PostTG_01.,
15     MoreBonus=as.numeric.df.PostTG_02.,
16     Responsible=as.numeric.df.PostTG_03.,
17     ProtectOthers=as.numeric.df.PostTG1_09.,
18      AllReceive=as.numeric.df.PostTG_11.,
19     TGscore=Scale1
20   )
21
22 p <- corData %>% ggpairs(
23   mapping = ggplot2::aes(colour=cond),
24   lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
         position_jitter()),
25                discrete = "blank", combo="dot"),
26   diag = list(discrete="barDiag", combo="denstrip",
27               continuous = wrap("densityDiag", alpha=0.3 )),
28   upper = list(combo = wrap("box_no_facet", alpha=0.5),
29                continuous = wrap("cor", size=4))) +
30   theme(panel.grid.major = element_blank()
31
32   )
```

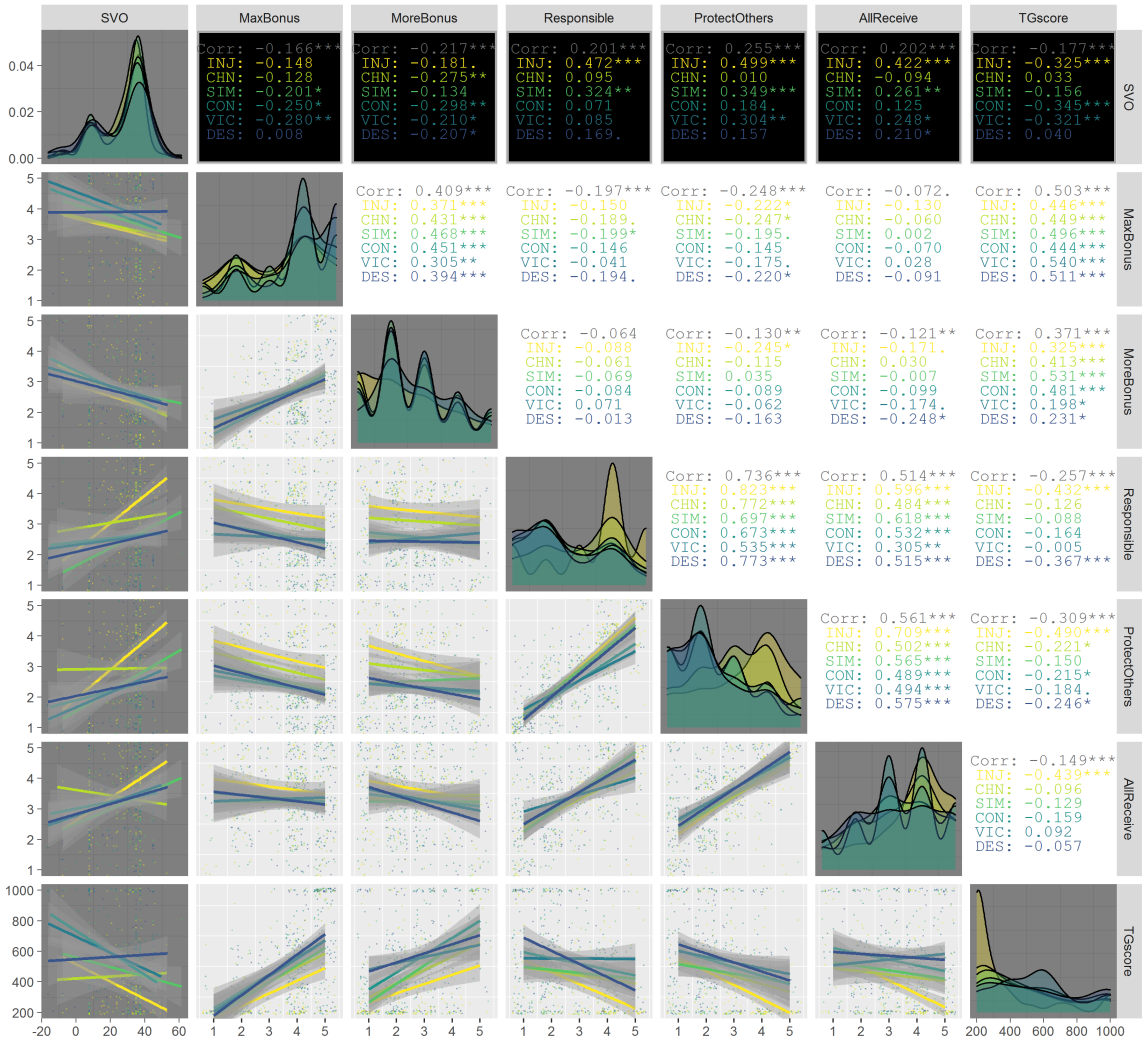

Figure SA2-23
Motivation and SVO

SA2-5.4 Motivation and CRT

```
1
2 corData=data.frame(
3   CRT=scoresCRT$CRTscore,
4   as.numeric(df$PostTG_02),
5   as.numeric(df$PostTG_04),
6   as.numeric(df$PostTG_07),
7   scoresTG
8 )
9
10 corData <- corData %>%
11   rename(
12     MoreBonus=as.numeric.df.PostTG_02.,
13     MakeSwitch=as.numeric.df.PostTG_04.,
```

```
    FollowGut=as.numeric.df.PostTG_07.,
    TGscore=Scale1
  )

p <- corData %>% ggpairs(
  mapping = ggplot2::aes(colour=cond),
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
               discrete = "blank", combo="dot"),
  diag = list(discrete="barDiag", combo="denstrip",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()

  )
```

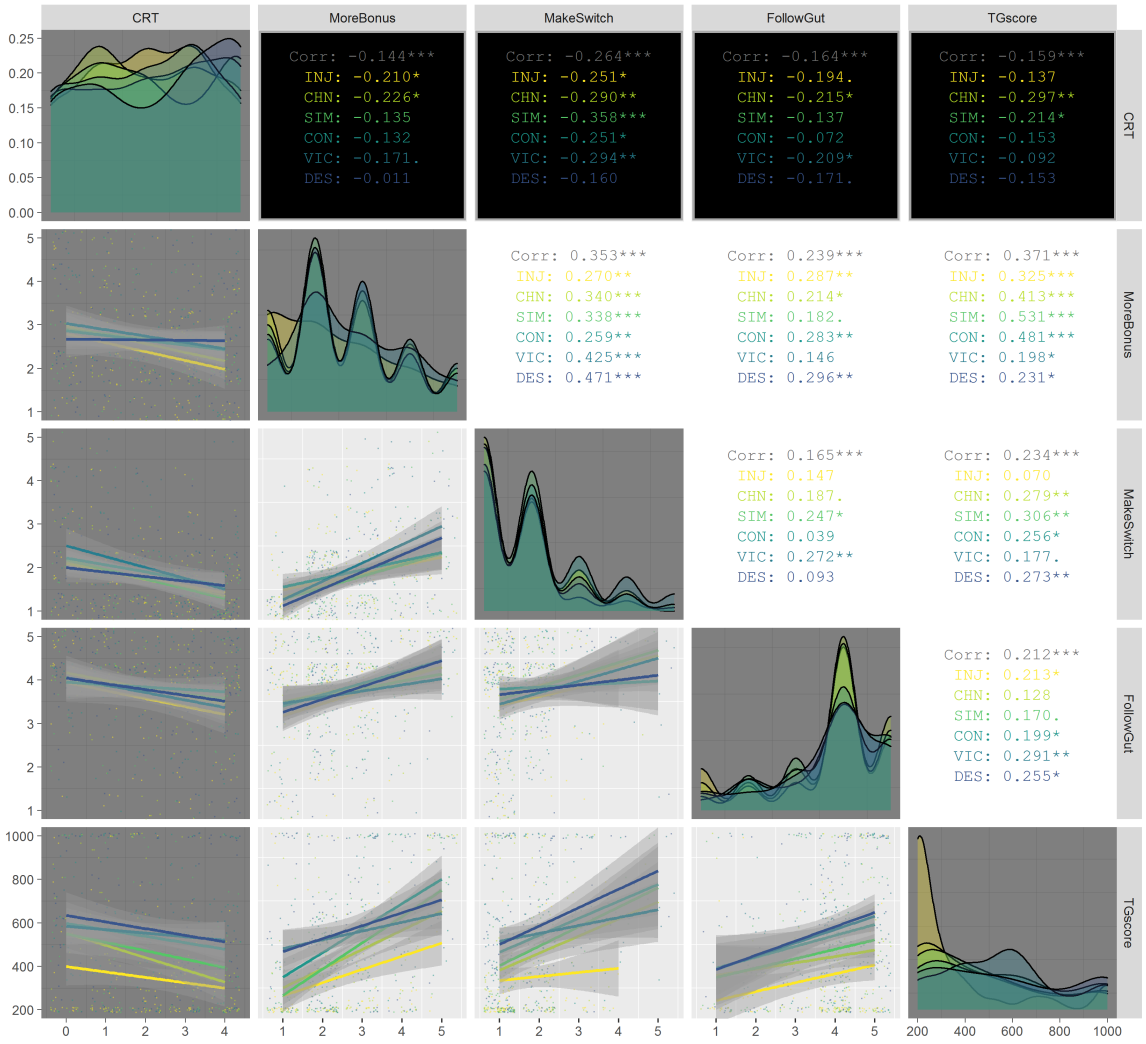

Figure SA2-24
Motivation and CRT

SA2-5.5 Specific postquestionnaires and game behavior

All postquestionnaires in the five intervention conditions shared the same structure. Dropping the additional question for the simulator condition (highly correlated with the "understood" item and unique to one condition), all condition-specific variables are joined in single variables and correlated with the game scores.

```
postAlldf=data.frame(
  Post1S=as.numeric(df$AddPostquestSimul_1),
  Post1I=as.numeric(df$AddPostInjunctive_1),
  Post1D=as.numeric(df$AddPostDescriptive_1),
  Post1V=as.numeric(df$AddPostVicar_1),
  Post1C=as.numeric(df$AddPostquestChain_1),
  Post2S=as.numeric(df$AddPostquestSimul_2),
  Post2I=as.numeric(df$AddPostInjunctive_2),
```

```
  Post2D=as.numeric(df$AddPostDescriptive_2),
  Post2V=as.numeric(df$AddPostVicar_2),
  Post2C=as.numeric(df$AddPostquestChain_2),
  Post3S=as.numeric(df$AddPostquestSimul_4),
  Post3I=as.numeric(df$AddPostInjunctive_3),
  Post3D=as.numeric(df$AddPostDescriptive_3),
  Post3V=as.numeric(df$AddPostVicar_3),
  Post3C=as.numeric(df$AddPostquestChain_3),
  Post4S=as.numeric(df$AddPostquestSimul_5),
  Post4I=as.numeric(df$AddPostInjunctive_4),
  Post4D=as.numeric(df$AddPostDescriptive_4),
  Post4V=as.numeric(df$AddPostVicar_4),
  Post4C=as.numeric(df$AddPostquestChain_4),
  Post5S=as.numeric(df$AddPostquestSimul_6),
  Post5I=as.numeric(df$AddPostInjunctive_5),
  Post5D=as.numeric(df$AddPostDescriptive_5),
  Post5V=as.numeric(df$AddPostVicar_5),
  Post5C=as.numeric(df$AddPostquestChain_5),
  Post6S=as.numeric(df$AddPostquestSimul_7),
  Post6I=as.numeric(df$AddPostInjunctive_6),
  Post6D=as.numeric(df$AddPostDescriptive_6),
  Post6V=as.numeric(df$AddPostVicar_6),
  Post6C=as.numeric(df$AddPostquestChain_6)
)

postAlldf[is.na(postAlldf)] <- 0

postAllJointdf=data.frame(
  scoresTG,
  Liked=postAlldf$Post1S+postAlldf$Post1I+
    postAlldf$Post1D+postAlldf$Post1V+postAlldf$Post1C,
  Understood=postAlldf$Post2S+postAlldf$Post2I+
    postAlldf$Post2D+postAlldf$Post2V+postAlldf$Post2C,
  Influenced=postAlldf$Post3S+postAlldf$Post3I+
    postAlldf$Post3D+postAlldf$Post3V+postAlldf$Post3C,
  WastedTime=postAlldf$Post4S+postAlldf$Post4I+
    postAlldf$Post4D+postAlldf$Post4V+postAlldf$Post4C,
  ReducedH=postAlldf$Post5S+postAlldf$Post5I+
    postAlldf$Post5D+postAlldf$Post5V+postAlldf$Post5C,
  ReducedG=postAlldf$Post6S+postAlldf$Post6I+
    postAlldf$Post6D+postAlldf$Post6V+postAlldf$Post6C
)

postAllJointdf <- postAllJointdf %>%
  rename(
    TGscore=Scale1
  )%>%filter(df$conditionShortName !="Control")

condFiltered = data.frame(
  cond
)
condFiltered <-  filter(condFiltered,df$conditionShortName !="Control")

p <- postAllJointdf %>% ggpairs(
  mapping = ggplot2::aes(colour=condFiltered$cond),
```

```
  lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1, position=
      position_jitter()),
                discrete = "blank", combo="dot"),
  diag = list(discrete="barDiag", combo="denstrip",
              continuous = wrap("densityDiag", alpha=0.3 )),
  upper = list(combo = wrap("box_no_facet", alpha=0.5),
               continuous = wrap("cor", size=4))) +
  theme(panel.grid.major = element_blank()
  )
```

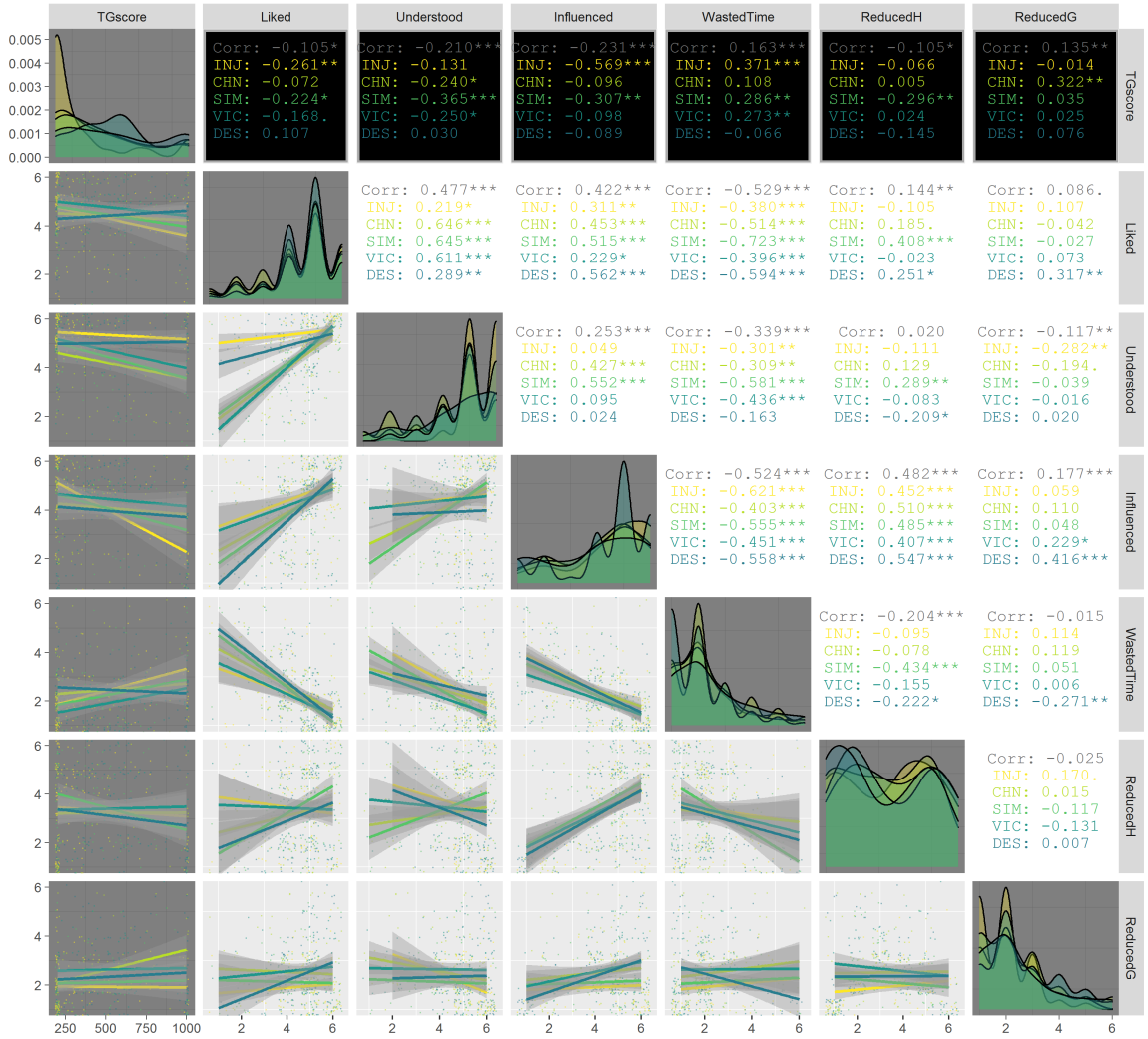

Figure SA2-25

Reactions to interventions and game scores

SA2-5.6 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)
```

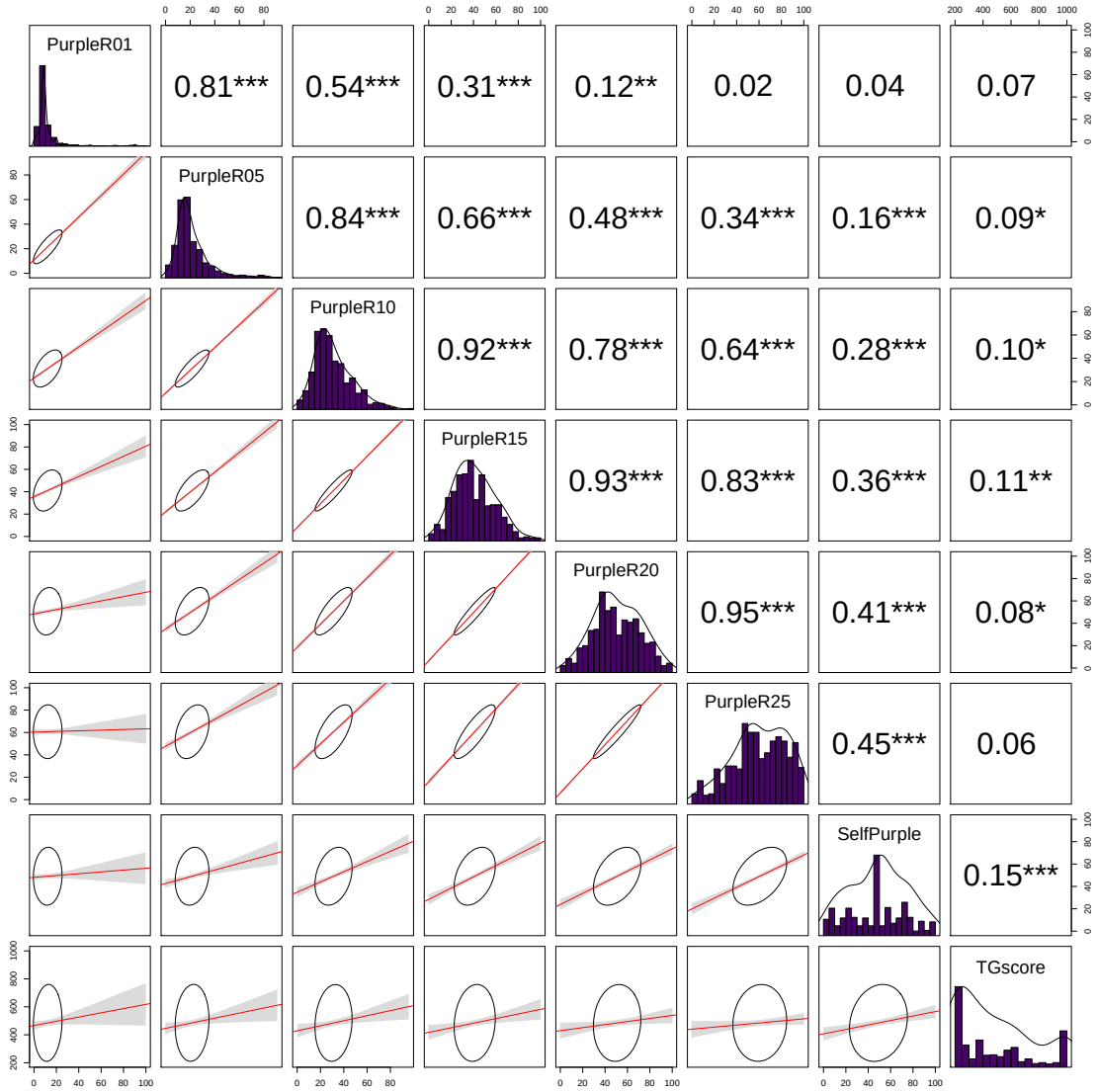

Figure SA2-26

Postquestionnaire expectation percent purple: Correlation with game score

```
round<-c()
round[1:600]<-1
round[601:1200]<-5
round[1201:1800]<-10
round[1801:2400]<-15
round[2401:3000]<-20
round[3001:3600]<-25

participantID<-c()
participantID[1:600]<-1:600
participantID[601:1200]<-1:600
participantID[1201:1800]<-1:600
participantID[1801:2400]<-1:600
participantID[2401:3000]<-1:600
participantID[3001:3600]<-1:600

condround<-c()
condround[1:600]<-df$conditionShortName
condround[601:1200]<-df$conditionShortName
condround[1201:1800]<-df$conditionShortName
condround[1801:2400]<-df$conditionShortName
condround[2401:3000]<-df$conditionShortName
condround[3001:3600]<-df$conditionShortName

estimatePurple<-c()
estimatePurple[1:600] <-df$PostTGestimatepurple_1
estimatePurple[601:1200] <-df$PostTGestimatepurple_2
estimatePurple[1201:1800] <-df$PostTGestimatepurple_3
estimatePurple[1801:2400] <-df$PostTGestimatepurple_4
estimatePurple[2401:3000] <-df$PostTGestimatepurple_5
estimatePurple[3001:3600] <-df$PostTGestimatepurple_6

lineFramePurple<-data.frame(
  cond=as.factor(condround),
  round,
  estimatePurple,
  participantID
)

library(nlme)

##
## Attaching package:  'nlme'
## The following object is masked from 'package:dplyr':
##
```

```
##     collapse

ggplot(lineFramePurple, aes(x = round, y = estimatePurple, color = cond, group=cond) ) +
  geom_point(position = "jitter") +
  geom_smooth(se = TRUE,size=2)+
  theme_minimal()+
  scale_colour_manual(values = c("#56C667FF","#39558CFF","#287D8EFF",
             "#FDE725FF","#B8DE29FF","#1F968BFF"),
             labels=c("CON","CHN","DES","INJ","SIM","VIC"))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

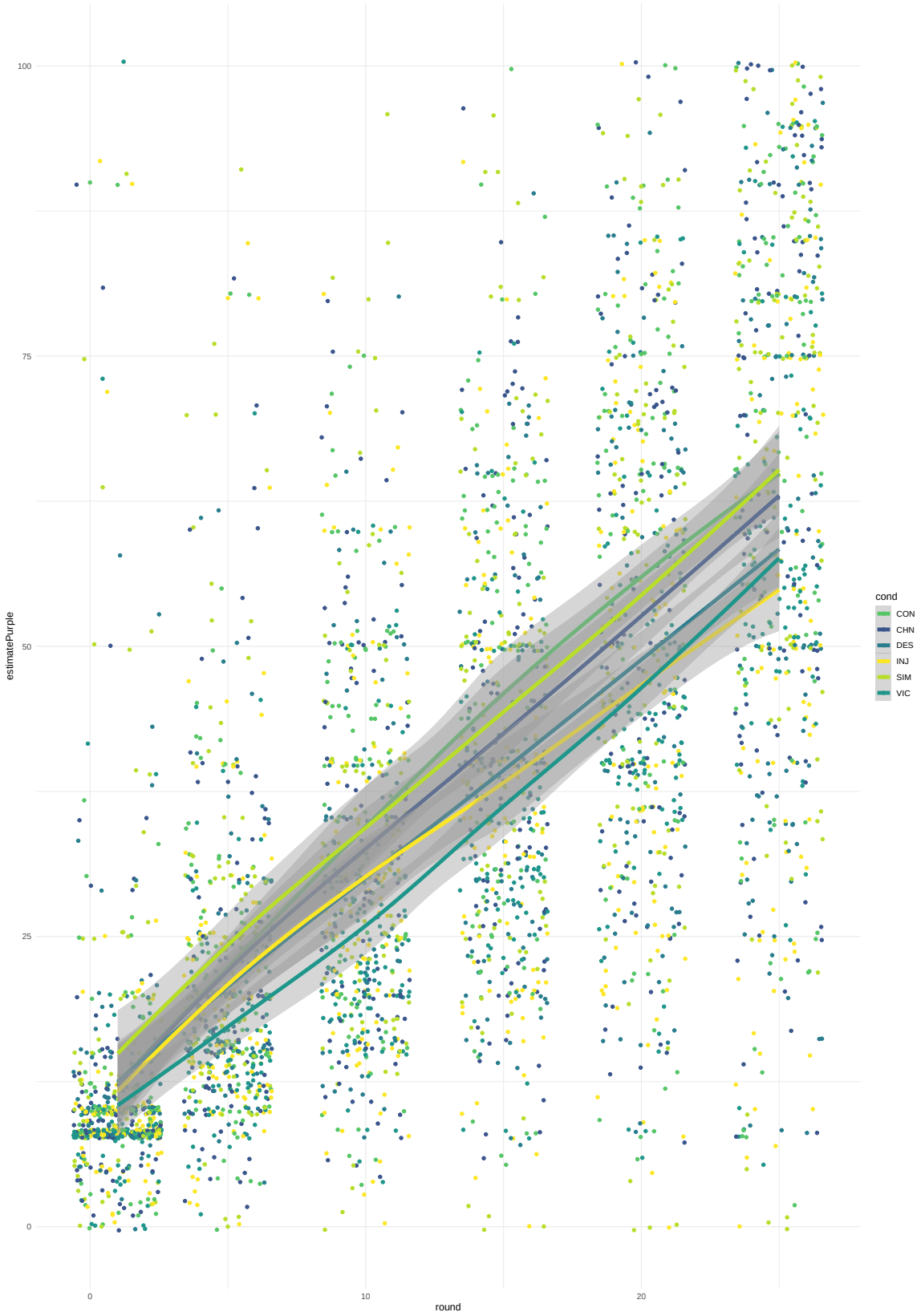

Figure SA2-27

Postquestionnaire expectation percent purple: Estimates by group

```
resAov <- aov(df$PostTGestimatepurple_1 ~ df$conditionShortName)
summary(resAov)

##                          Df Sum Sq Mean Sq F value Pr(>F)
## df$conditionShortName     5   1060   212.1    1.29  0.266
## Residuals               594  97625   164.3

resAov <- aov(df$PostTGestimatepurple_2 ~ df$conditionShortName)
summary(resAov)

##                          Df Sum Sq Mean Sq F value Pr(>F)
## df$conditionShortName     5   2766   553.2   2.975 0.0116 *
## Residuals               594 110448   185.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimatepurple_2 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff        lwr       upr     p adj
## Chain-Control              0.22  -5.293438  5.733438 0.9999973
## Descriptive-Control       -1.34  -6.853438  4.173438 0.9825009
## Injunctive-Control        -1.05  -6.563438  4.463438 0.9942784
## Simulator-Control          2.48  -3.033438  7.993438 0.7927117
## Vicarious-Control         -4.66 -10.173438  0.853438 0.1521441
## Descriptive-Chain         -1.56  -7.073438  3.953438 0.9658746
## Injunctive-Chain          -1.27  -6.783438  4.243438 0.9862574
## Simulator-Chain            2.26  -3.253438  7.773438 0.8501235
## Vicarious-Chain           -4.88 -10.393438  0.633438 0.1169308
## Injunctive-Descriptive     0.29  -5.223438  5.803438 0.9999893
## Simulator-Descriptive      3.82  -1.693438  9.333438 0.3545229
## Vicarious-Descriptive     -3.32  -8.833438  2.193438 0.5179432
## Simulator-Injunctive       3.53  -1.983438  9.043438 0.4468275
## Vicarious-Injunctive      -3.61  -9.123438  1.903438 0.4205276
## Vicarious-Simulator       -7.14 -12.653438 -1.626562 0.0031777

resAov <- aov(df$PostTGestimatepurple_3 ~ df$conditionShortName)
summary(resAov)

##                          Df Sum Sq Mean Sq F value  Pr(>F)
## df$conditionShortName     5   5309  1061.8   4.294 0.00076 ***
```

```
## Residuals             594 146864   247.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimatepurple_3 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                           diff        lwr        upr     p adj
## Chain-Control            -2.01  -8.367715  4.3477153 0.9454788
## Descriptive-Control      -4.47 -10.827715  1.8877153 0.3376776
## Injunctive-Control       -4.40 -10.757715  1.9577153 0.3558241
## Simulator-Control        -0.21  -6.567715  6.1477153 0.9999989
## Vicarious-Control        -8.63 -14.987715 -2.2722847 0.0016092
## Descriptive-Chain        -2.46  -8.817715  3.8977153 0.8787973
## Injunctive-Chain         -2.39  -8.747715  3.9677153 0.8913296
## Simulator-Chain           1.80  -4.557715  8.1577153 0.9657834
## Vicarious-Chain          -6.62 -12.977715 -0.2622847 0.0356934
## Injunctive-Descriptive    0.07  -6.287715  6.4277153 1.0000000
## Simulator-Descriptive     4.26  -2.097715 10.6177153 0.3934586
## Vicarious-Descriptive    -4.16 -10.517715  2.1977153 0.4213183
## Simulator-Injunctive      4.19  -2.167715 10.5477153 0.4128825
## Vicarious-Injunctive     -4.23 -10.587715  2.1277153 0.4017371
## Vicarious-Simulator      -8.42 -14.777715 -2.0622847 0.0023164

resAov <- aov(df$PostTGestimatepurple_4 ~ df$conditionShortName)
summary(resAov)

##                          Df Sum Sq Mean Sq F value   Pr(>F)
## df$conditionShortName     5   7075  1414.9   4.296 0.000758 ***
## Residuals               594 195651   329.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimatepurple_4 ~ df$conditionShortName)
##
## $`df$conditionShortName`
```

```
##                           diff        lwr        upr     p adj
## Chain-Control          -3.45 -10.788105  3.8881049 0.7601357
## Descriptive-Control    -6.61 -13.948105  0.7281049 0.1049679
## Injunctive-Control     -7.67 -15.008105 -0.3318951 0.0345231
## Simulator-Control      -1.58  -8.918105  5.7581049 0.9898967
## Vicarious-Control      -9.70 -17.038105 -2.3618951 0.0023805
## Descriptive-Chain      -3.16 -10.498105  4.1781049 0.8214560
## Injunctive-Chain       -4.22 -11.558105  3.1181049 0.5694743
## Simulator-Chain         1.87  -5.468105  9.2081049 0.9783906
## Vicarious-Chain        -6.25 -13.588105  1.0881049 0.1459151
## Injunctive-Descriptive -1.06  -8.398105  6.2781049 0.9984604
## Simulator-Descriptive   5.03  -2.308105 12.3681049 0.3669389
## Vicarious-Descriptive  -3.09 -10.428105  4.2481049 0.8349731
## Simulator-Injunctive    6.09  -1.248105 13.4281049 0.1675423
## Vicarious-Injunctive   -2.03  -9.368105  5.3081049 0.9690383
## Vicarious-Simulator    -8.12 -15.458105 -0.7818951 0.0202529
```

```
resAov <- aov(df$PostTGestimatepurple_5 ~ df$conditionShortName)
summary(resAov)
```

```
##                        Df Sum Sq Mean Sq F value  Pr(>F)
## df$conditionShortName   5   8267  1653.4   3.755 0.00235 **
## Residuals             594 261571   440.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(resAov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimatepurple_5 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                           diff       lwr        upr     p adj
## Chain-Control          -3.34 -11.82473  5.1447299 0.8707640
## Descriptive-Control    -7.03 -15.51473  1.4547299 0.1689720
## Injunctive-Control     -9.15 -17.63473 -0.6652701 0.0259834
## Simulator-Control      -1.70 -10.18473  6.7847299 0.9927523
## Vicarious-Control      -9.78 -18.26473 -1.2952701 0.0132547
## Descriptive-Chain      -3.69 -12.17473  4.7947299 0.8152342
## Injunctive-Chain       -5.81 -14.29473  2.6747299 0.3681314
## Simulator-Chain         1.64  -6.84473 10.1247299 0.9938682
## Vicarious-Chain        -6.44 -14.92473  2.0447299 0.2534186
## Injunctive-Descriptive -2.12 -10.60473  6.3647299 0.9802008
```

```
## Simulator-Descriptive   5.33  -3.15473 13.8147299 0.4690796
## Vicarious-Descriptive  -2.75 -11.23473  5.7347299 0.9396002
## Simulator-Injunctive    7.45  -1.03473 15.9347299 0.1226573
## Vicarious-Injunctive   -0.63  -9.11473  7.8547299 0.9999408
## Vicarious-Simulator    -8.08 -16.56473  0.4047299 0.0723080

resAov <- aov(df$PostTGestimatepurple_6 ~ df$conditionShortName)
summary(resAov)

##                          Df Sum Sq Mean Sq F value  Pr(>F)
## df$conditionShortName   5   9225  1845.1   3.249 0.00666 **
## Residuals             594 337351   567.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimatepurple_6 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff         lwr        upr     p adj
## Chain-Control             -1.89 -11.5257201  7.7457201 0.9934343
## Descriptive-Control       -6.53 -16.1657201  3.1057201 0.3802184
## Injunctive-Control       -10.02 -19.6557201 -0.3842799 0.0361059
## Simulator-Control          0.38  -9.2557201 10.0157201 0.9999975
## Vicarious-Control         -7.20 -16.8357201  2.4357201 0.2700159
## Descriptive-Chain         -4.64 -14.2757201  4.9957201 0.7410441
## Injunctive-Chain          -8.13 -17.7657201  1.5057201 0.1535793
## Simulator-Chain            2.27  -7.3657201 11.9057201 0.9847884
## Vicarious-Chain           -5.31 -14.9457201  4.3257201 0.6150701
## Injunctive-Descriptive    -3.49 -13.1257201  6.1457201 0.9058369
## Simulator-Descriptive      6.91  -2.7257201 16.5457201 0.3152426
## Vicarious-Descriptive     -0.67 -10.3057201  8.9657201 0.9999572
## Simulator-Injunctive      10.40   0.7642799 20.0357201 0.0257780
## Vicarious-Injunctive       2.82  -6.8157201 12.4557201 0.9605715
## Vicarious-Simulator       -7.58 -17.2157201  2.0557201 0.2170768
```

SA2-5.6.1 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)
```

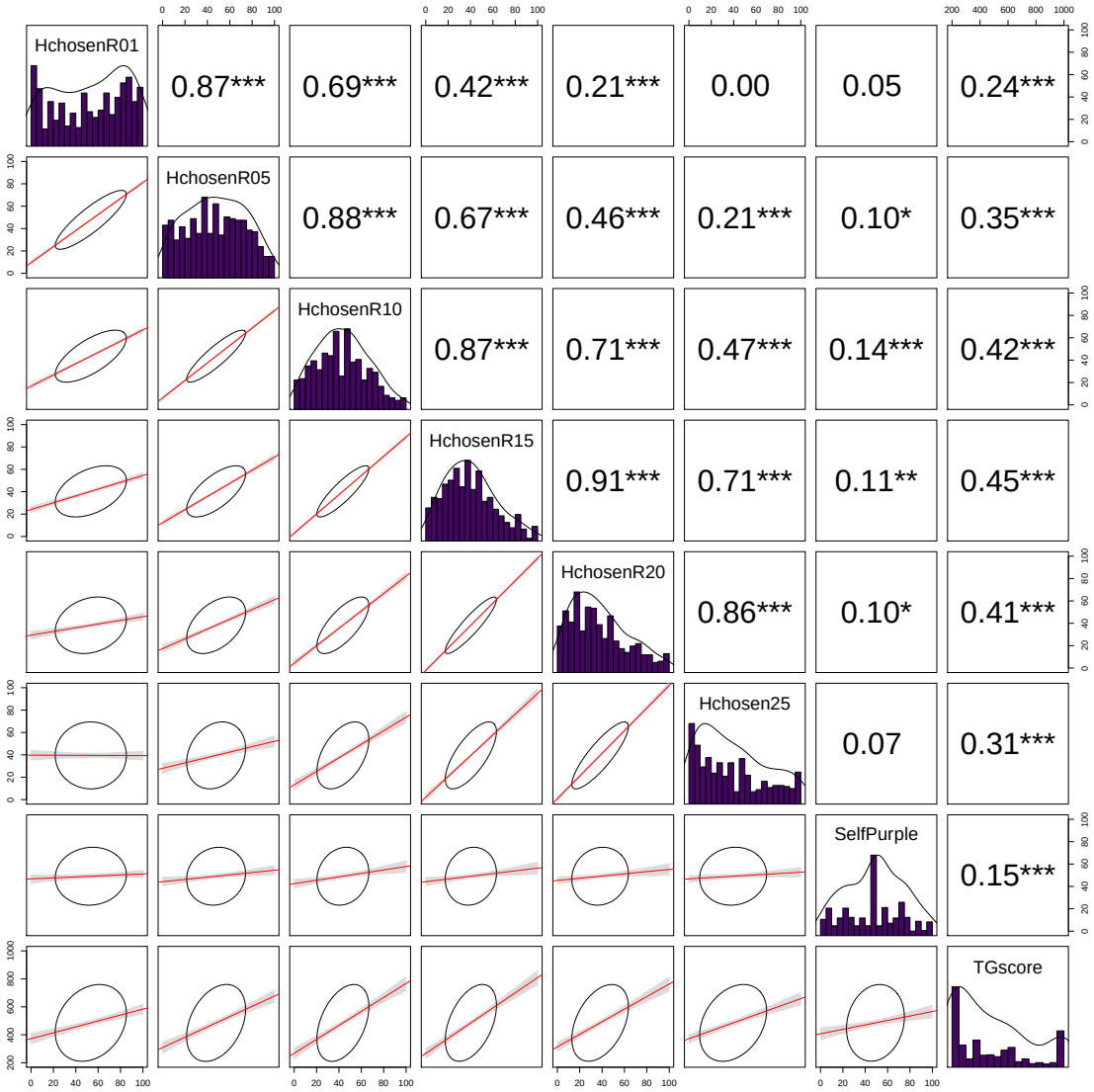

Figure SA2-28

Postquestionnaire expectation percent H-choices: Correlation with game score

```
estimateH<-c()
estimateH[1:600] <-df$PostTGestimateH_1
estimateH[601:1200] <-df$PostTGestimateH_2
estimateH[1201:1800] <-df$PostTGestimateH_3
estimateH[1801:2400] <-df$PostTGestimateH_4
estimateH[2401:3000] <-df$PostTGestimateH_5
estimateH[3001:3600] <-df$PostTGestimateH_6

lineFrameH<-data.frame(
  cond=as.factor(condround),
  round,
  estimateH,
  participantID,
  scoresTG
)

ggplot(lineFrameH, aes(x = round, y = estimateH, color = cond, group=cond) ) +
  geom_point(position = "jitter") +
  geom_smooth(se = TRUE,size=2)+
  theme_minimal()+
  scale_colour_manual(values = c("#56C667FF","#39558CFF","#287D8EFF",
           "#FDE725FF","#B8DE29FF", "#1F968BFF"),
           labels=c("CON","CHN","DES","INJ","SIM","VIC"))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

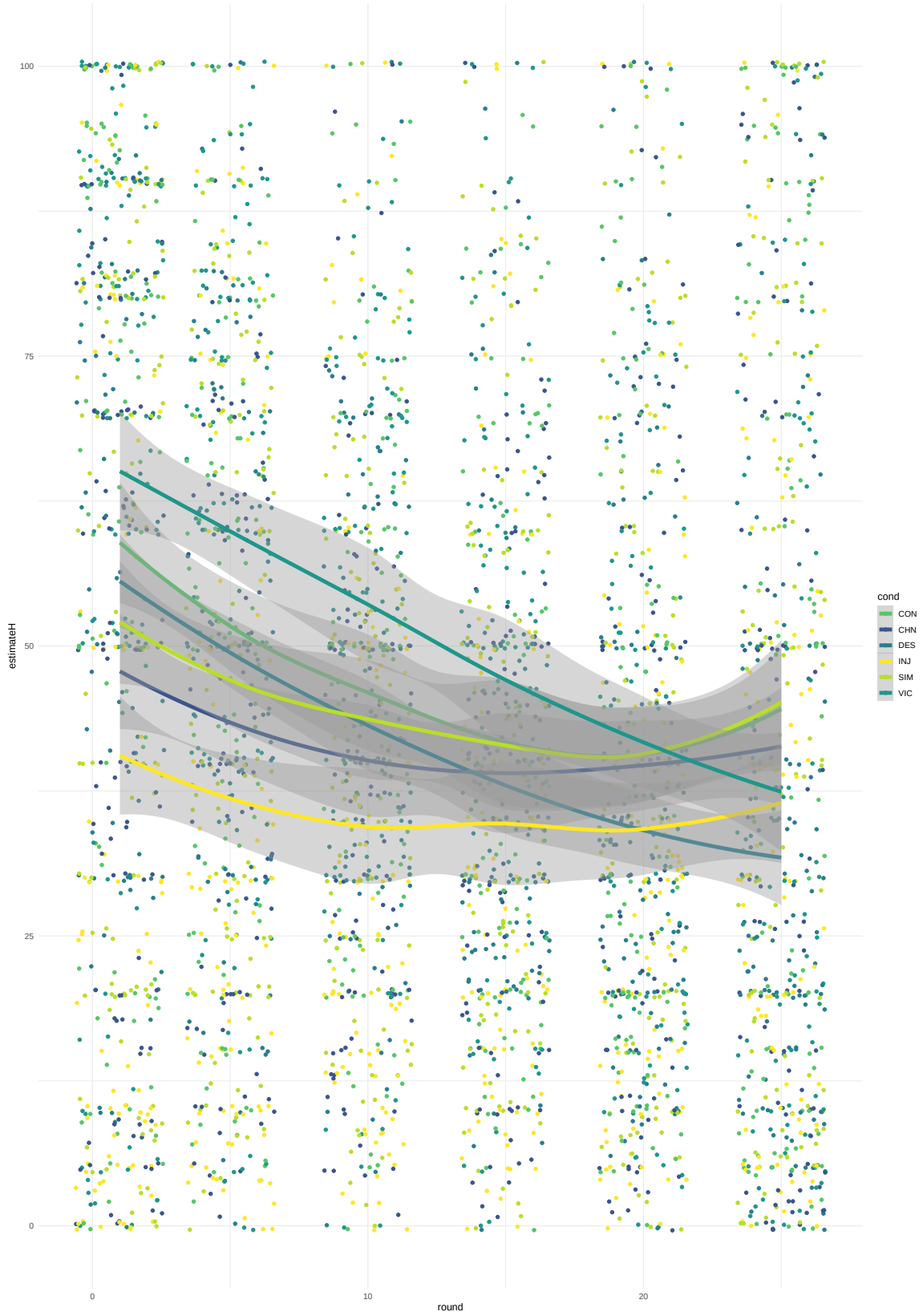

Figure SA2-29

Postquestionnaire expectation percent H-choices: Estimates by group

```r
resAov <- aov(df$PostTGestimateH_1 ~ df$conditionShortName)
summary(resAov)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## df$conditionShortName    5  37278    7456   7.774 4.24e-07 ***
## Residuals              594 569672     959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(resAov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_1 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff         lwr       upr     p adj
## Chain-Control           -11.29 -23.8114727  1.231473 0.1043303
## Descriptive-Control      -3.51 -16.0314727  9.011473 0.9672183
## Injunctive-Control      -18.61 -31.1314727 -6.088527 0.0003571
## Simulator-Control        -6.88 -19.4014727  5.641473 0.6181256
## Vicarious-Control         5.97  -6.5514727 18.491473 0.7490903
## Descriptive-Chain         7.78  -4.7414727 20.301473 0.4818501
## Injunctive-Chain         -7.32 -19.8414727  5.201473 0.5513413
## Simulator-Chain           4.41  -8.1114727 16.931473 0.9156205
## Vicarious-Chain          17.26   4.7385273 29.781473 0.0012702
## Injunctive-Descriptive -15.10 -27.6214727 -2.578527 0.0079302
## Simulator-Descriptive    -3.37 -15.8914727  9.151473 0.9725260
## Vicarious-Descriptive     9.48  -3.0414727 22.001473 0.2560648
## Simulator-Injunctive     11.73  -0.7914727 24.251473 0.0811233
## Vicarious-Injunctive     24.58  12.0585273 37.101473 0.0000005
## Vicarious-Simulator      12.85   0.3285273 25.371473 0.0404298
```

```r
resAov <- aov(df$PostTGestimateH_2 ~ df$conditionShortName)
summary(resAov)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## df$conditionShortName    5  29660    5932   9.124 2.25e-08 ***
## Residuals              594 386179     650
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(resAov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_2 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff         lwr       upr     p adj
## Chain-Control             -7.76 -18.0694938  2.549494 0.2622151
## Descriptive-Control       -1.78 -12.0894938  8.529494 0.9963926
## Injunctive-Control       -14.14 -24.4494938 -3.830506 0.0013730
## Simulator-Control         -4.63 -14.9394938  5.679494 0.7938105
## Vicarious-Control          8.71  -1.5994938 19.019494 0.1524896
## Descriptive-Chain          5.98  -4.3294938 16.289494 0.5600075
## Injunctive-Chain          -6.38 -16.6894938  3.929494 0.4865011
## Simulator-Chain            3.13  -7.1794938 13.439494 0.9539510
## Vicarious-Chain           16.47   6.1605062 26.779494 0.0000876
## Injunctive-Descriptive   -12.36 -22.6694938 -2.050506 0.0084989
## Simulator-Descriptive     -2.85 -13.1594938  7.459494 0.9691322
## Vicarious-Descriptive     10.49   0.1805062 20.799494 0.0434230
## Simulator-Injunctive       9.51  -0.7994938 19.819494 0.0900324
## Vicarious-Injunctive      22.85  12.5405062 33.159494 0.0000000
## Vicarious-Simulator       13.34   3.0305062 23.649494 0.0032139

resAov <- aov(df$PostTGestimateH_3 ~ df$conditionShortName)
summary(resAov)

##                         Df Sum Sq Mean Sq F value   Pr(>F)
## df$conditionShortName    5  20304    4061   7.954 2.87e-07 ***
## Residuals              594 303271     511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_3 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff         lwr        upr     p adj
## Chain-Control             -5.96 -15.0960439  3.1760439 0.4248443
## Descriptive-Control       -2.93 -12.0660439  6.2060439 0.9421626
## Injunctive-Control       -11.71 -20.8460439 -2.5739561 0.0036576
## Simulator-Control         -2.30 -11.4360439  6.8360439 0.9795265
```

```
## Vicarious-Control        7.49  -1.6460439 16.6260439 0.1782944
## Descriptive-Chain        3.03  -6.1060439 12.1660439 0.9336611
## Injunctive-Chain        -5.75 -14.8860439  3.3860439 0.4668754
## Simulator-Chain          3.66  -5.4760439 12.7960439 0.8621302
## Vicarious-Chain         13.45   4.3139561 22.5860439 0.0004236
## Injunctive-Descriptive  -8.78 -17.9160439  0.3560439 0.0677019
## Simulator-Descriptive    0.63  -8.5060439  9.7660439 0.9999589
## Vicarious-Descriptive   10.42   1.2839561 19.5560439 0.0148453
## Simulator-Injunctive     9.41   0.2739561 18.5460439 0.0391995
## Vicarious-Injunctive    19.20  10.0639561 28.3360439 0.0000000
## Vicarious-Simulator      9.79   0.6539561 18.9260439 0.0275717

resAov <- aov(df$PostTGestimateH_4 ~ df$conditionShortName)
summary(resAov)

##                         Df Sum Sq Mean Sq F value  Pr(>F)
## df$conditionShortName   5   8765  1753.0   3.414 0.00475 **
## Residuals             594 305043   513.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(resAov)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_4 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                           diff          lwr       upr     p adj
## Chain-Control            -2.67 -11.83270303  6.492703 0.9612906
## Descriptive-Control      -3.73 -12.89270303  5.432703 0.8538261
## Injunctive-Control       -7.03 -16.19270303  2.132703 0.2422196
## Simulator-Control        -0.28  -9.44270303  8.882703 0.9999993
## Vicarious-Control         5.36  -3.80270303 14.522703 0.5506088
## Descriptive-Chain        -1.06 -10.22270303  8.102703 0.9994753
## Injunctive-Chain         -4.36 -13.52270303  4.802703 0.7506657
## Simulator-Chain           2.39  -6.77270303 11.552703 0.9760499
## Vicarious-Chain           8.03  -1.13270303 17.192703 0.1240417
## Injunctive-Descriptive   -3.30 -12.46270303  5.862703 0.9078849
## Simulator-Descriptive     3.45  -5.71270303 12.612703 0.8906613
## Vicarious-Descriptive     9.09  -0.07270303 18.252703 0.0532475
## Simulator-Injunctive      6.75  -2.41270303 15.912703 0.2853911
## Vicarious-Injunctive     12.39   3.22729697 21.552703 0.0017049
## Vicarious-Simulator       5.64  -3.52270303 14.802703 0.4927065
```

```r
resAov <- aov(df$PostTGestimateH_5 ~ df$conditionShortName)
summary(resAov)
```

```
##                         Df Sum Sq Mean Sq F value Pr(>F)
## df$conditionShortName    5   6232  1246.4   1.964 0.0822 .
## Residuals              594 376969   634.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(resAov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_5 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                                   diff        lwr       upr     p adj
## Chain-Control          -4.000000e-01 -10.585811  9.785811 0.9999975
## Descriptive-Control    -6.250000e+00 -16.435811  3.935811 0.4963536
## Injunctive-Control     -6.860000e+00 -17.045811  3.325811 0.3874796
## Simulator-Control      -4.000000e-01 -10.585811  9.785811 0.9999975
## Vicarious-Control       1.400000e+00  -8.785811 11.585811 0.9987884
## Descriptive-Chain      -5.850000e+00 -16.035811  4.335811 0.5709073
## Injunctive-Chain       -6.460000e+00 -16.645811  3.725811 0.4579237
## Simulator-Chain        -2.131628e-14 -10.185811 10.185811 1.0000000
## Vicarious-Chain         1.800000e+00  -8.385811 11.985811 0.9959739
## Injunctive-Descriptive -6.100000e-01 -10.795811  9.575811 0.9999796
## Simulator-Descriptive   5.850000e+00  -4.335811 16.035811 0.5709073
## Vicarious-Descriptive   7.650000e+00  -2.535811 17.835811 0.2645608
## Simulator-Injunctive    6.460000e+00  -3.725811 16.645811 0.4579237
## Vicarious-Injunctive    8.260000e+00  -1.925811 18.445811 0.1881804
## Vicarious-Simulator     1.800000e+00  -8.385811 11.985811 0.9959739
```

```r
resAov <- aov(df$PostTGestimateH_6 ~ df$conditionShortName)
summary(resAov)
```

```
##                         Df Sum Sq Mean Sq F value  Pr(>F)
## df$conditionShortName    5  13584  2716.8   3.082 0.00935 **
## Residuals              594 523686   881.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(resAov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$PostTGestimateH_6 ~ df$conditionShortName)
##
## $`df$conditionShortName`
##                            diff        lwr        upr     p adj
## Chain-Control             -3.37 -15.375446  8.6354462 0.9670224
## Descriptive-Control      -12.88 -24.885446 -0.8745538 0.0272708
## Injunctive-Control        -7.98 -19.985446  4.0254462 0.4028566
## Simulator-Control          0.69 -11.315446 12.6954462 0.9999834
## Vicarious-Control         -7.22 -19.225446  4.7854462 0.5194049
## Descriptive-Chain         -9.51 -21.515446  2.4954462 0.2103017
## Injunctive-Chain          -4.61 -16.615446  7.3954462 0.8822203
## Simulator-Chain            4.06  -7.945446 16.0654462 0.9282270
## Vicarious-Chain           -3.85 -15.855446  8.1554462 0.9421774
## Injunctive-Descriptive     4.90  -7.105446 16.9054462 0.8524375
## Simulator-Descriptive     13.57   1.564554 25.5754462 0.0163165
## Vicarious-Descriptive      5.66  -6.345446 17.6654462 0.7579795
## Simulator-Injunctive       8.67  -3.335446 20.6754462 0.3073771
## Vicarious-Injunctive       0.76 -11.245446 12.7654462 0.9999731
## Vicarious-Simulator       -7.91 -19.915446  4.0954462 0.4131927
```

SA2-6 Exploratory Analyses

SA2-6.1 COVID-19 related scales and predictor variables

```
corData=data.frame(
  scaleCVW$scores,
  scaleCVCP$scores,
  scaleCVTO$scores,
  CRT=scoresCRT$CRTscore,
  risk=as.numeric(df$RTGeneral),
  SVO=scoresSVO$angleSVO
)


corData <- corData %>%
  rename(
    Worries=sclCOVWorry,
    Compliance=sclCompliance,
    Tradeoffs=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="blue",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

SA2-6.2 COVID-19 related scales and HEXACO-scales

```
corData=data.frame(
  scaleCVW$scores,
  scaleCVCP$scores,
  scaleCVTO$scores,
  scoresHEXACO$scores
)


corData <- corData %>%
  rename(
    Worries=sclCOVWorry,
    Compliance=sclCompliance,
    Tradeoffs=sclTradeoffs,
    H60=scaleH60,
    E60=scaleE60,
    Xs=scaleBX,
    As=scaleBA,
    Cs=scaleBC,
    Os=scaleBO,
  )

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)
```

SA2-6.3 COVID-19 related scales and politics

```
corData=data.frame(
  scaleCVW$scores,
  scaleCVCP$scores,
  scaleCVTO$scores,
  scaleSECS$scores,
```

```
  PrefTrump=df$polCandScale_2,
  PrefBiden=df$polCandScale_4,
  Conservatism=df$polPosition
  )


corData <- corData %>%
  rename(
    Worries=sclCOVWorry,
    Compliance=sclCompliance,
    Tradeoffs=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="blue",show.points=FALSE,rug=FALSE,cex.cor=1,wt=NULL,
             stars=TRUE,ci=TRUE,alpha=.05)
```

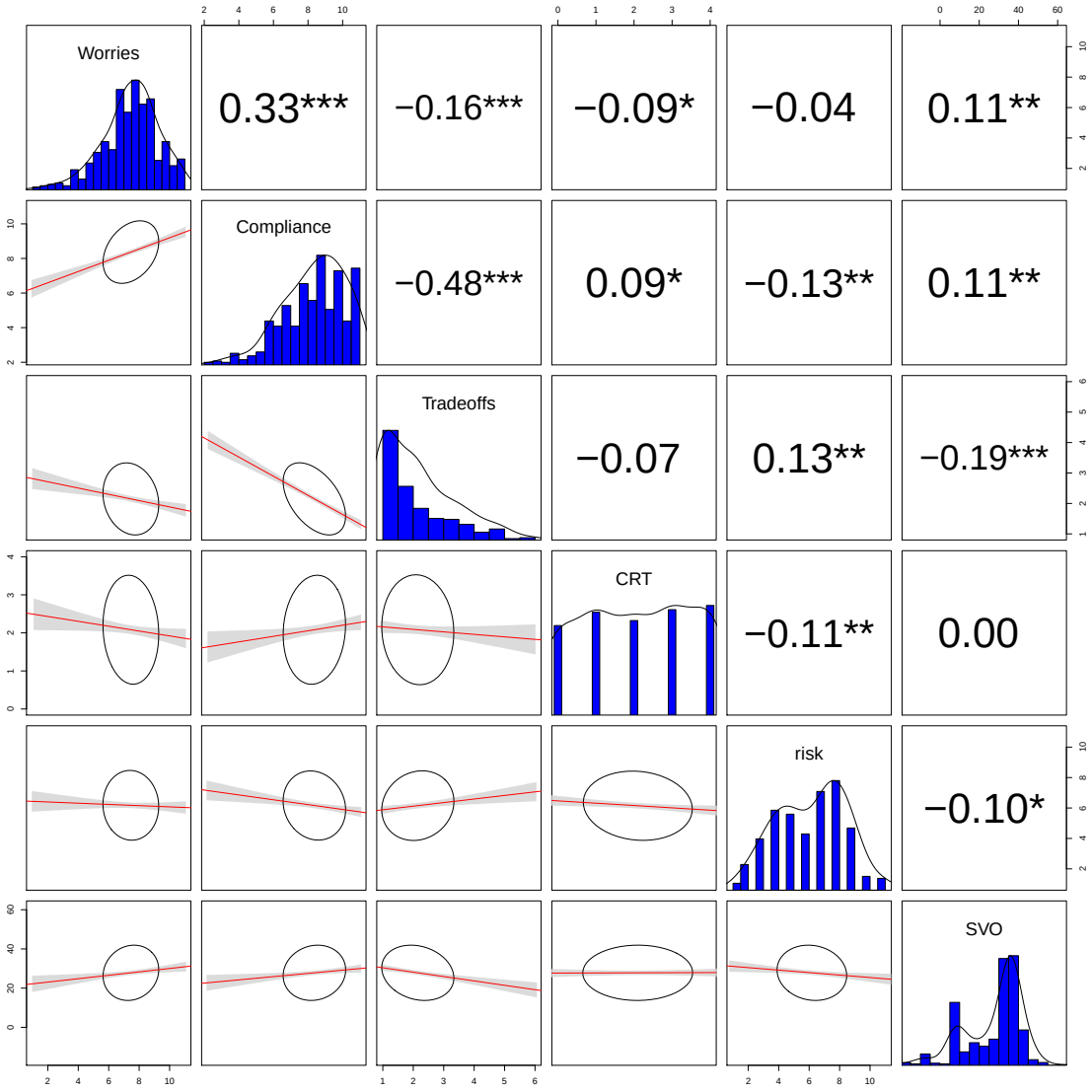

Figure SA2-30
COVID-19 related scales and predictors

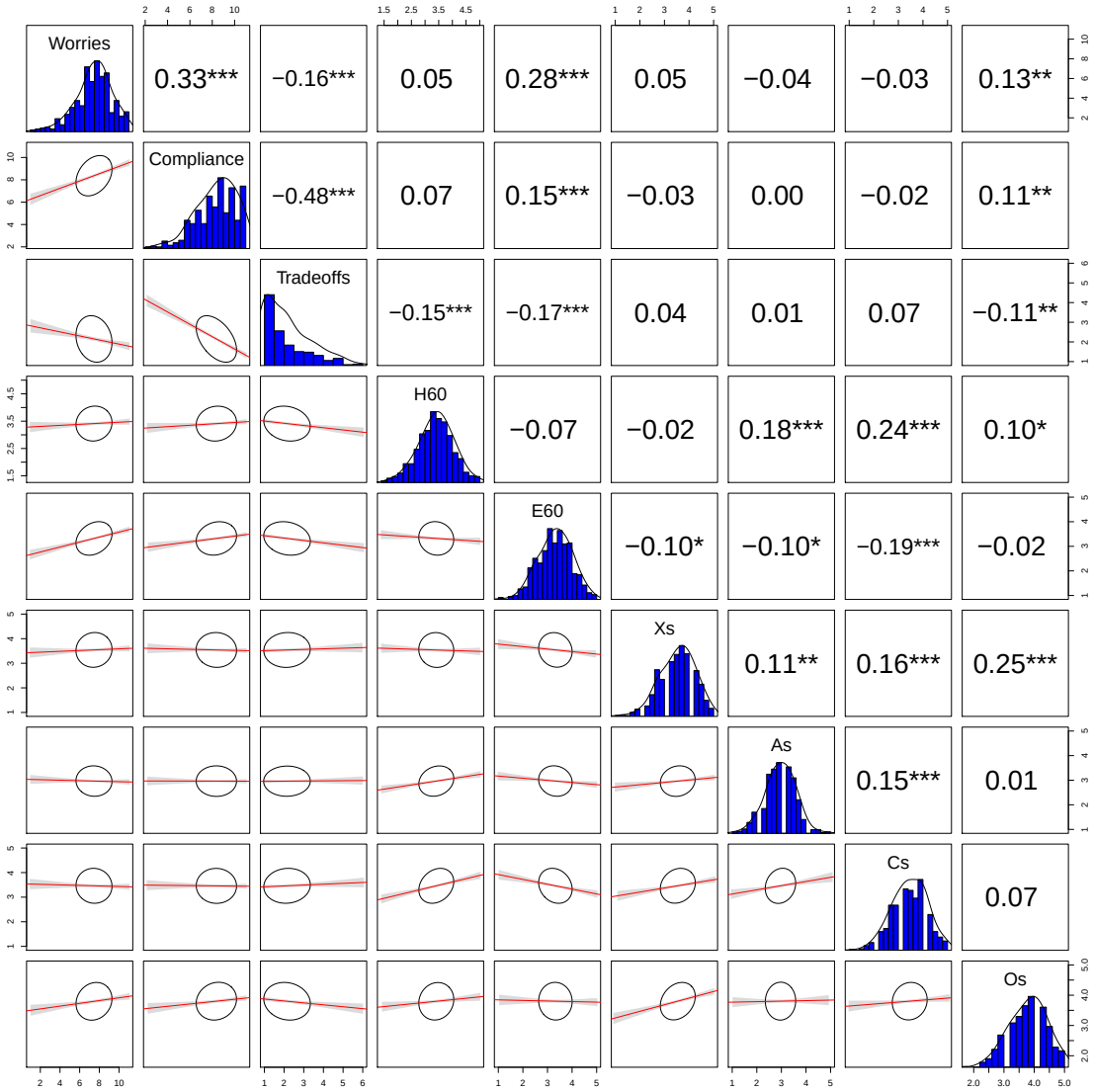

Figure SA2-31
COVID-19 related scales and HEXACO scales

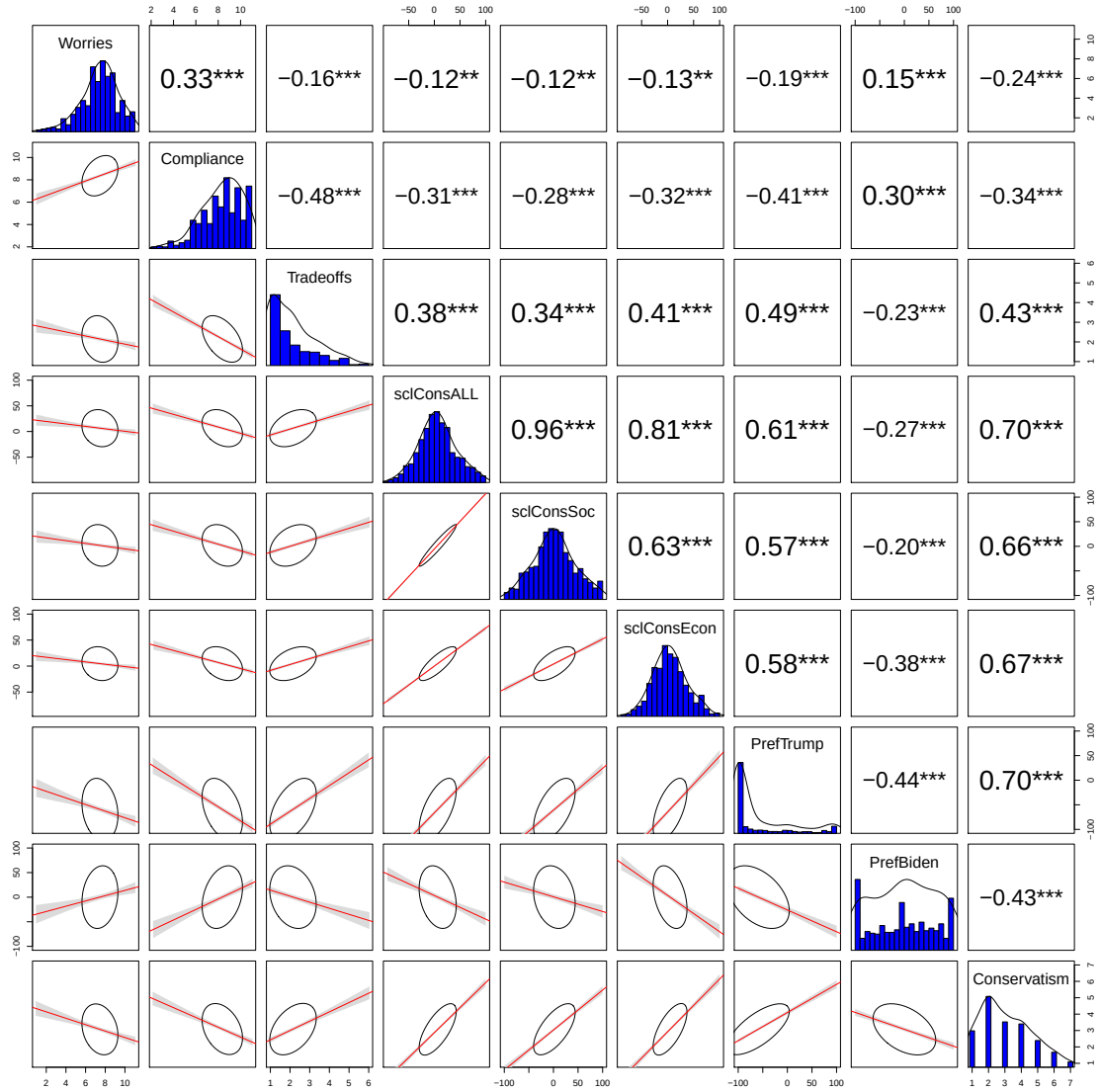

Figure SA2-32
COVID-19 related scales and politics

```
ggplot(data=corData,aes(x=Conservatism,y=Compliance))+
  geom_boxplot()+
  theme_dark()
```

```
ggplot(data=corData,aes(x=PrefTrump,y=Compliance))+
  geom_point()+
  geom_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
ggplot(data=corData,aes(x=PrefBiden,y=Compliance))+
  geom_point()+
  geom_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
ggplot(data=corData,aes(x=Conservatism,y=Tradeoffs))+
  geom_boxplot()
```

```
ggplot(data=corData,aes(x=PrefTrump,y=Tradeoffs))+
  geom_point()+
  geom_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
ggplot(data=corData,aes(x=Conservatism,y=Worries))+
  geom_boxplot()
```

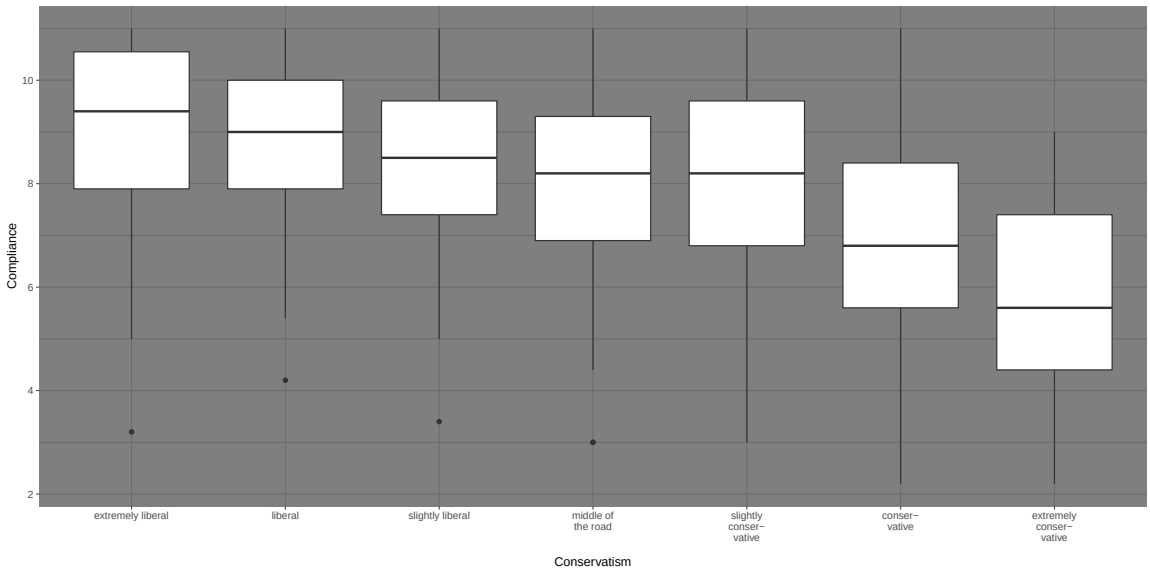

Figure SA2-33
Conservatism and Compliance

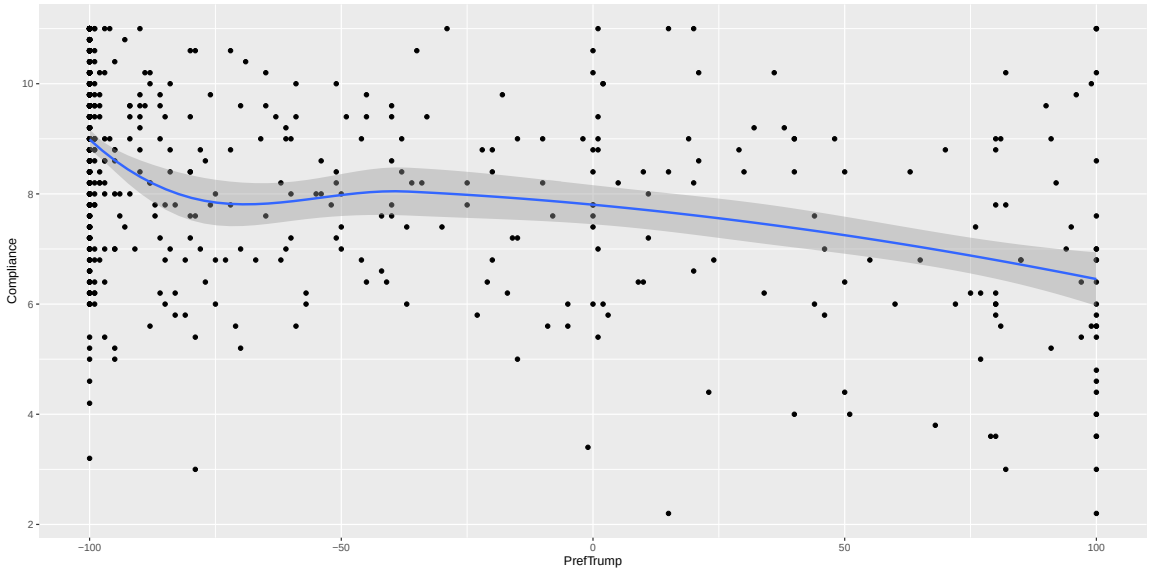

Figure SA2-34
Preference for Trump and Compliance

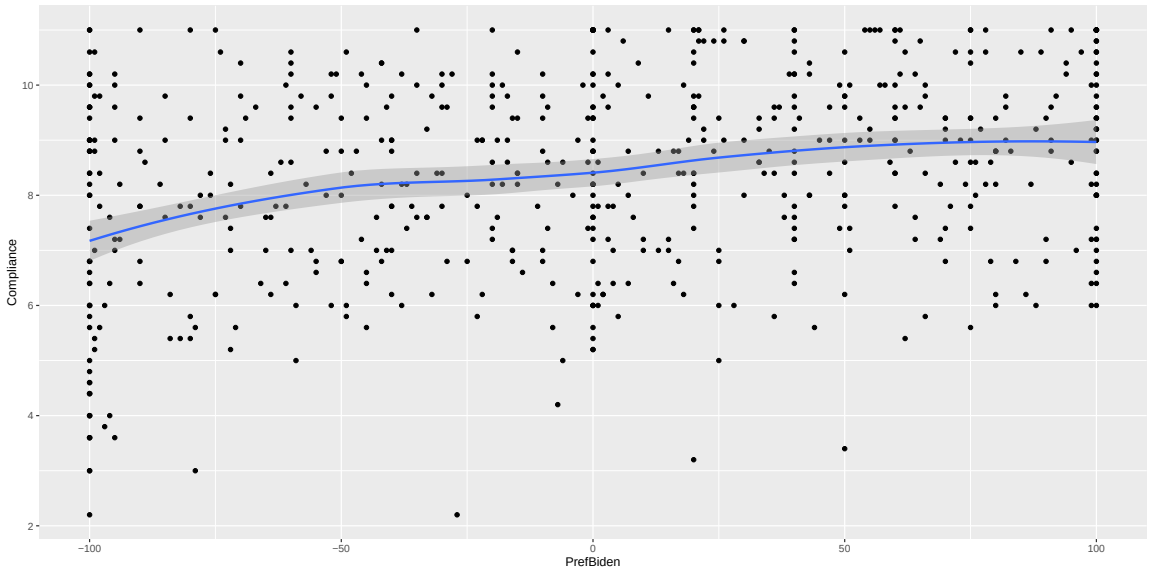

Figure SA2-35
Preference for Biden and Compliance

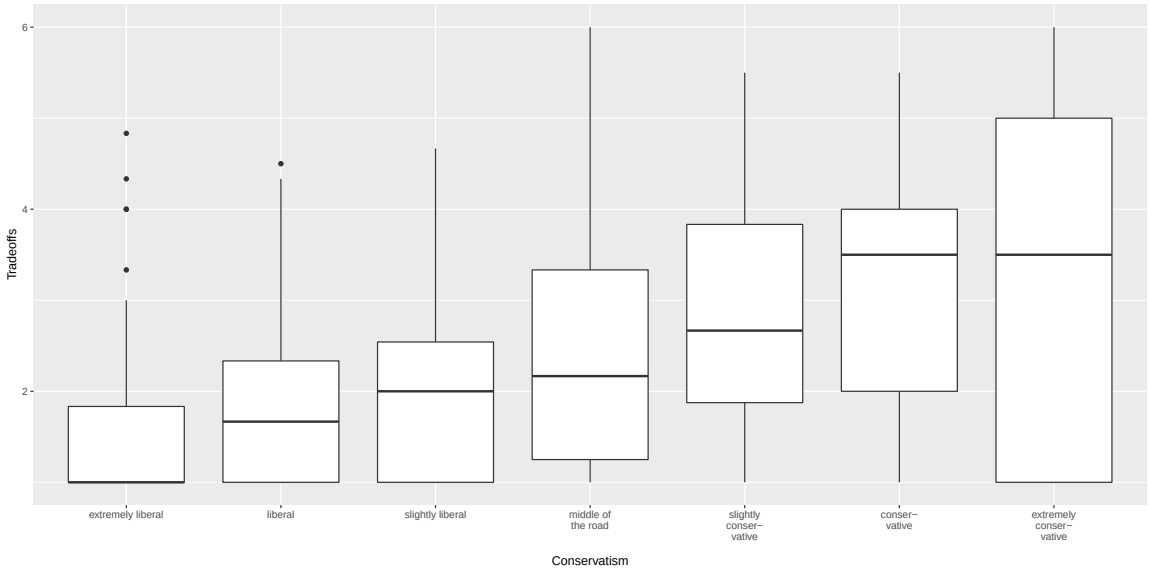

Figure SA2-36
Conservatism and Tradeoffs

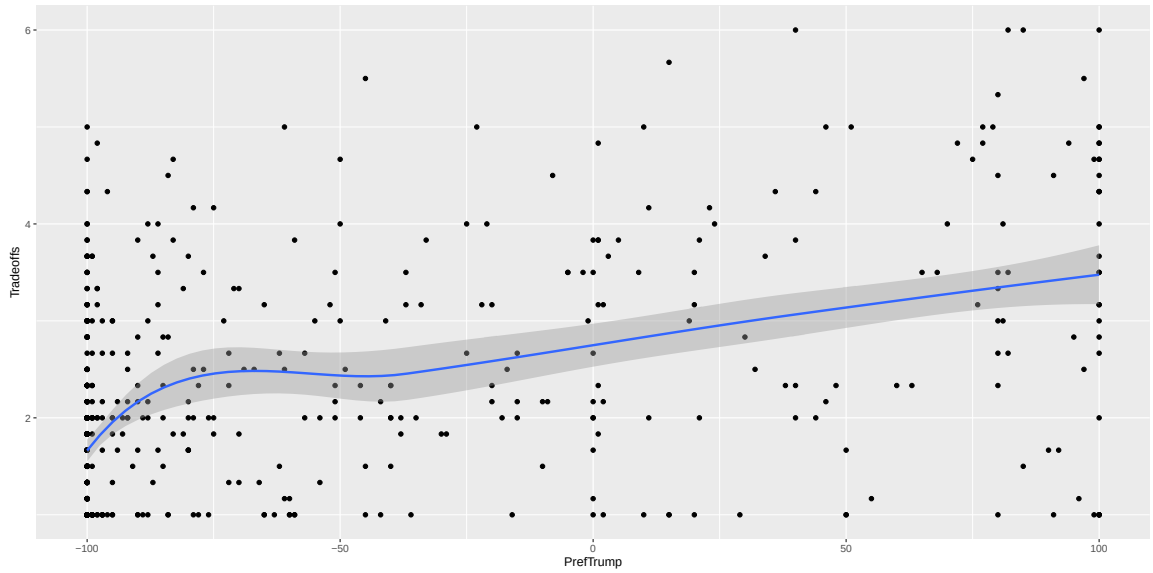

Figure SA2-37
Preference for Trump and Treadeoffs

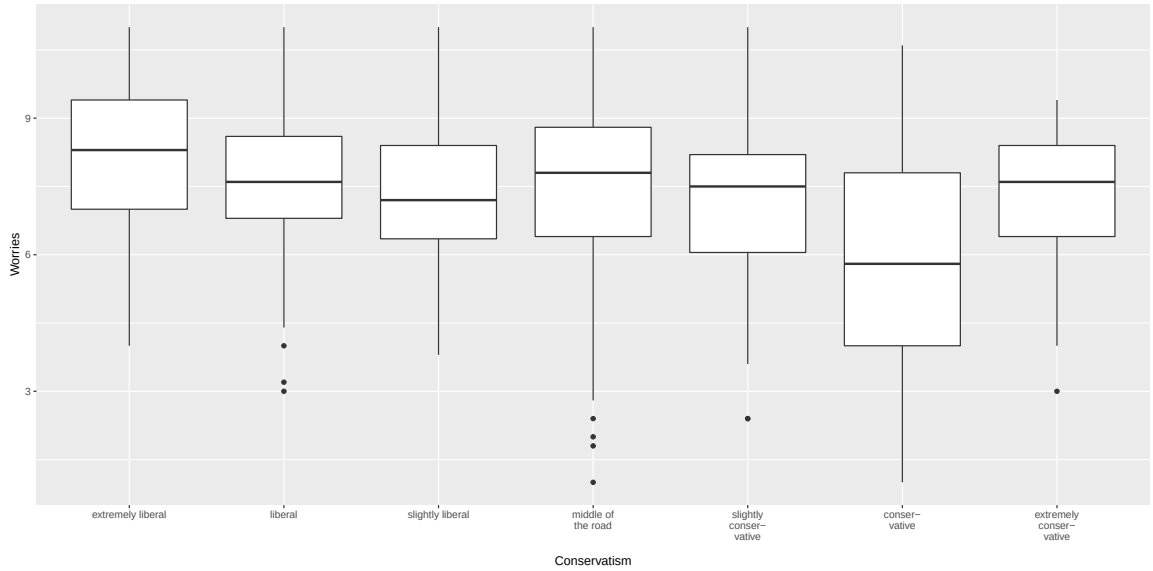

Figure SA2-38
Conservatism and Worries

SA2-6.4 COVID-19 testing and game results

```
corData=data.frame(
 scoresTG,
df$COV_PosTest
)

corData <- corData %>%
  rename(
    TGscore=Scale1,
    PositiveTest=df.COV_PosTest
  )

resAOV<-aov(corData$TGscore~corData$PositiveTest)
summary(resAOV)

##                         Df   Sum Sq Mean Sq F value Pr(>F)
## corData$PositiveTest   2   892938  446469   6.011 0.0026 **
## Residuals            597 44343689   74278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ggplot(data=corData,aes(x=PositiveTest,y=TGscore))+
  geom_boxplot()
```

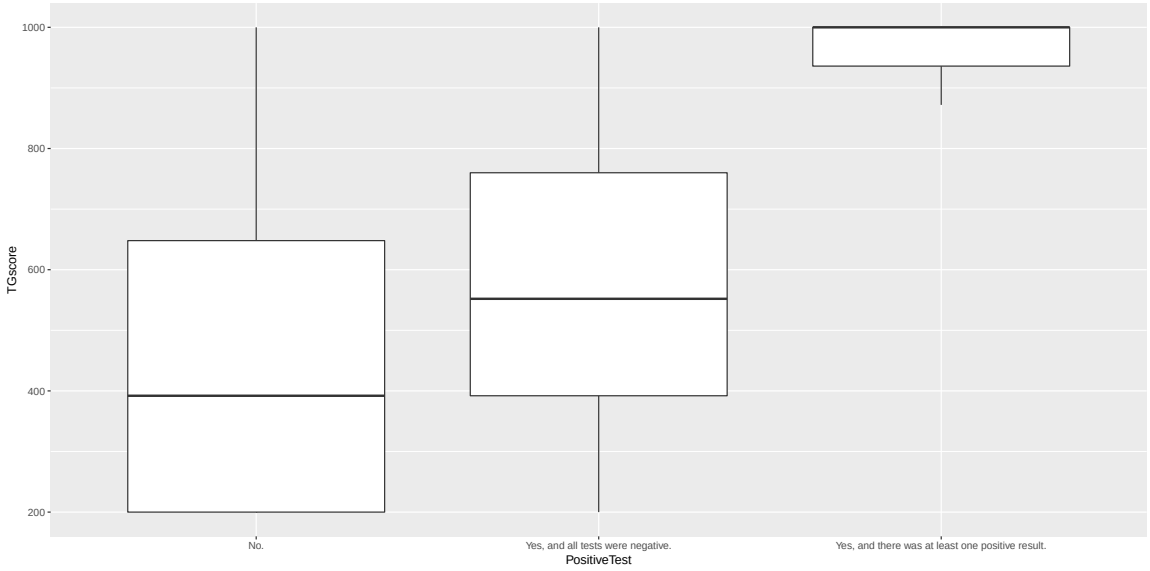

Figure SA2-39
COVID-19 testing and game results