

Trait Sexual Desire-Linked Subjective Sexual Arousal to Erotic and Non-Erotic Stimuli: Gender, Relationship Status, and Gender-Specificity

Code and analyses

Milena Vásquez-Amézquita ^{1,2,3,4,*}

Juan David Leongómez ^{2,4}

Marina Begoña Martínez-González ¹

Meredith L. Chivers ⁵

11 February, 2025

¹ Departamento de Ciencias Sociales, Universidad de la Costa, Barranquilla, 080002, Colombia.

² Facultad de Psicología, Universidad El Bosque, Bogotá, 110121, Colombia.

³ Grupo de Investigación en Cultura, Educación y Sociedad, Universidad de la Costa, Barranquilla, 080002, Colombia.

⁴ CODEC: Ciencias Cognitivas y del Comportamiento, Universidad El Bosque, Bogotá, 110121, Colombia.

⁵ Department of Psychology, Queen's University, Kingston ON K7L 3N6, Canada.

* Correspondence: mvasquezam@unbosque.edu.co

Description

This document contains all code, and step by step explanations for all analyses, figures and tables (including supplementary figures and tables) for:

Vásquez-Amézquita, M., Leongómez, J. D., Martínez-González, M. B., & Chivers, M. L. (in prep).
Trait Sexual Desire-Linked Subjective Sexual Arousal to Erotic and Non-Erotic Stimuli: Gender, Relationship Status, and Gender-Specificity

Data available from the Open Science Framework (OSF): <https://doi.org/10.17605/OSF.IO/3V2E7>.
All analyses were planned by Milena Vásquez-Amézquita and Juan David Leongómez. This document and its underlying code were created in R Markdown by Juan David Leongómez using \LaTeX .

Contents

1 Preliminaries	2
1.1 Load packages	2
1.2 Define color palettes	3
1.3 Custom functions	3
1.3.1 <code>pval.lev</code>	3
1.3.2 <code>pval.stars</code>	3
1.3.3 <code>corr.stars</code>	3
1.3.4 <code>summary.sig</code> and <code>summary.sig.boot</code>	4
1.3.5 <code>emms.sig</code>	6
1.3.6 <code>contr.stars</code>	9
1.3.7 <code>prob.dist.tab</code>	9
1.4 Load and wrangle data	10

2	Descriptives	12
2.0.1	Figure S1. Demographic characteristics of the sample	12
2.1	Descriptive statistics of the participants by gender	13
2.1.1	Table S1. Descriptive statistics of the participants by gender	14
2.1.2	Figure S2. Distribution of participants' measured variables by gender	15
2.2	Correlations between measured variables	17
2.2.1	Table S2. Correlations between measured variables	18
2.3	Internal consistency	20
2.3.1	Table S3. Internal consistency of construct variables	21
3	Controlling the effects of Relationship Duration and MGSS sexual satisfaction (Partner) on sexual desire dimensions	22
4	Hypothesis tests	26
4.1	Hypothesis 1: All dimensions of trait sexual desire (TSD) will be higher in men than in women, and the differences will be stronger or weaker according to relationship status	26
4.1.1	Data	26
4.1.2	Hypothesis 1a: Solitary TSD	27
4.1.2.1	Model the effects of relationship type and gender on Solitary TSD	27
4.1.2.1.1	Model assumptions	27
5	Session info (for reproducibility)	30
6	Supplementary references	31

1 Preliminaries

1.1 Load packages

This file was created using `knitr` (Xie, 2014), mostly using `tidyverse` (Wickham et al., 2019) syntax. As such, data wrangling was mainly done using packages such as `dplyr` (Wickham et al., 2023), and most figures were created or modified using `ggplot2` (Wickham, 2016). Tables were created using `knitr::kable` and `kableExtra` (Zhu, 2021).

Linear mixed models were fitted using `lmerTest` (Kuznetsova et al., 2017), assumptions were performed using `performance` (Lüdtke et al., 2021), contrasts and interactions were explored using `emmeans` (Lenth, 2023), and interactions were investigated using the package `interactions` (Long, 2019).

All packages used in this file can be directly installed from the Comprehensive R Archive Network ([CRAN](https://cran.r-project.org/)). For a complete list of packages used to create this file, and their versions, see section 5, at the end of the document.

```
library(readxl)
library(ltm)
library(car)
library(tidyverse)
library(ggpubr)
library(tidyquant)
library(performance)
library(kableExtra)
library(psych)
library(scales)
library(lmerTest)
library(emmeans)
library(berryFunctions)
library(bestNormalize)
library(effectsize)
#library(dplyr)
#library(interactions)
```

```
#library(MetBrewer)
#library(ggpmisc)
#library(effectsize)
#library(rstatix)
```

1.2 Define color palettes

Individual color palettes for figures by gender, stimuli sex, or relationship type.

```
# Palette to color figures by gender
color.Gender <- c("#bd3106", "#5b7314")
# Palette to color figures by stimuli sex
color.StimuliSex <- c("#454b87", "#d9700e")
# Palette to color figures by relationship type
color.Relationship <- c("#b39e05", "#a513d6")
# Palette to color figures by dimension type
color.Dimension <- c("#582310", "#318f49", "#0cb4bb")
```

1.3 Custom functions

1.3.1 pval.lev

This function takes p-values and formats them in L^AT_EX, highlighting significant results in bold.

```
# Version 1 for LaTeX format
pval.lev <- function(pvals) {
  ifelse(pvals < 0.0001, "\\textbf{< 0.0001}",
    ifelse(pvals < 0.001, "\\textbf{< 0.001}",
      ifelse(pvals < 0.05, paste0("\\textbf{", round(pvals, 4), "}"),
        round(pvals, 2))))
}

# Version 2 without LaTeX format
pval.lev2 <- function(pvals) {
  ifelse(pvals < 0.0001, "< 0.0001",
    ifelse(pvals < 0.001, "< 0.001",
      ifelse(pvals < 0.05, paste0("= ", round(pvals, 4)),
        paste0("= ", round(pvals, 2))))
}
```

1.3.2 pval.stars

This function takes p-values and adds stars to represent significance levels.

```
pval.stars <- function(pvals) {
  ifelse(pvals < 0.0001, "****",
    ifelse(pvals < 0.001, "***",
      ifelse(pvals < 0.01, "**",
        ifelse(pvals < 0.05, "*", NA))))
}
```

1.3.3 corr.stars

This function creates a correlation matrix, and displays significance (function `corr.stars` modified from <http://myowelt.blogspot.com/2008/04/beautiful-correlation-tables-in-r.html>).

```
corr.stars <- function(x) {
  require(Hmisc)
```

```

x <- as.matrix(x)
R <- rcorr(x)$r
p <- rcorr(x)$P
# define notions for significance levels; spacing is important.
mystars <- ifelse(p < .001,
  paste0("\\textbf{", round(R, 2), "***}"),
  ifelse(p < .01,
    paste0("\\textbf{", round(R, 2), "**}"),
    ifelse(p < .05,
      paste0("\\textbf{", round(R, 2), "*}"),
      ifelse(p < .10,
        paste0(round(R, 2), "$^{\dagger}$"),
        format(round(R, 2), nsmall = 2))))))
# build a new matrix that includes the correlations with their appropriate stars
Rnew <- matrix(mystars,
  ncol = ncol(x))
diag(Rnew) <- paste(diag(R), " ",
  sep = "")
rownames(Rnew) <- colnames(x)
colnames(Rnew) <- paste(colnames(x), "",
  sep = "")
# remove upper triangle
Rnew <- as.matrix(Rnew)
Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
Rnew <- as.data.frame(Rnew)
# remove last column and return the matrix (which is now a data frame)
Rnew <- cbind(Rnew[1:length(Rnew) - 1])
return(Rnew)
}

```

1.3.4 summary.sig and summary.sig.boot

Functions to bold significant p values from summary model tables. It highlights significant p values, and formats the output in L^AT_EX, ready to be used with `kable`.

We used `summary` (regression-type tables of estimates) instead on ANOVA-type tables to display model results. This was because we needed to bootstrap estimates for the two models on Hypothesis 2 (see section ??). However, to obtain p -values that represent main effects and interactions, we used *sum-to-zero* contrasts (see e.g., Kaufman & Sweet, 1974; Keppel & Zedeck, 1989).

```

# Version 1 for models with no CIs
summary.sig <- function(mod, custom_caption) {
  modTab <- data.frame(summary(mod)$coefficients) |>
    rownames_to_column() |>
    mutate_at("rowname", str_replace_all, ":", " × ") |>
    mutate_at("rowname", str_replace_all, "~", "") |>
    mutate("rowname" = str_replace_all(rowname,
      c("Gender1" = "Gender [Women]",
        "Relationship1" = "Relationship [Stable]",
        Dimension1 = "Dimension [Attractive person DSD]",
        Dimension2 = "Dimension [Partner DSD]"))) |>
    select(rowname, Estimate, Std..Error, df, t.value, Pr...t..) |>
    mutate(Pr...t.. = pval.lev(Pr...t..)) |>
    kable(digits = 2,
      booktabs = TRUE,
      align = c("l", rep("c", 5)),
      linesep = "",

```

```

caption = custom_caption,
col.names = c("Effect", "Estimate", "Std. Error", "$df$", "$t$", "$p$"),
escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
footnote(general = paste0("Sexual desire was measured as the proportion of
each score divided by the maximum possible score
for each dimension.
 $R^2_{\text{conditional}}$  = ",
round(r2_nakagawa(mod)$R2_conditional, 3),
",  $R^2_{\text{marginal}}$  = ",
round(r2_nakagawa(mod)$R2_marginal, 3),
". Results are from linear mixed models for main
effects and interactions between sexual desire (SD) dimensions,
sex, and Stimuli sex.
Gender = participants gender (women, men);
Stimuli sex = sex of stimuli (female, male);
Attractive person DSD = Dyadic sexual desire toward an
attractive person;
Partner DSD = Dyadic sexual desire toward a partner.
\\\\\\textit{Sum-to-zero} contrasts were used to display
\\\\\\textit{p}-values that represent main effects and interactions
in an ANOVA-type manner (i.e. the intercept is the grand mean of
all cells, and estimates are differences between each category
mean and the mean of all categories).
\\\\\\textit{Single} was used as reference category
for relationship status, \\\\\\textit{Men} for gender,
and \\\\\\textit{Solitary} for sexual desire dimension.
Contrasted levels are in square brackets.
Significant effects are in bold."),
escape = FALSE,
threeparttable = TRUE,
footnote_as_chunk = TRUE)
return(modTab)
}

# Version 2 for models with bootstrap CIs
summary.sig.boot <- function(mod, modCI, custom_caption) {
  modTab <- left_join(data.frame(summary(mod)$coefficients) |>
    rownames_to_column(),
    data.frame(modCI) |>
    rownames_to_column(),
    by = "rowname") |>
  mutate_at("rowname", str_replace_all, ":", " × ") |>
  mutate_at("rowname", str_replace_all, "`", "") |>
  mutate("rowname" = str_replace_all(rowname,
    c("`Solitary sexual desire (C)`" =
      "Solitary SD (C)",
      "`Dyadic sexual desire (Attractive person) (C)`" =
        "Attractive person DSD (C)",
      "`Dyadic sexual desire (Partner) (C)`" =
        "Partner DSD (C)",
      "Relationship1" = "Relationship [Stable]",
      "Stimuli sex1" = "Stimuli sex [Female]",
      "Gender1" = "Gender [Women]")))) |>
  select(rowname, Estimate, X2.5., X97.5., Std..Error, df, t.value, Pr...t..) |>
  mutate(Pr...t.. = pval.lev(Pr...t..)) |>

```

```

kable(digits = 2,
      booktabs = TRUE,
      align = c("l", rep("c", 7)),
      linesep = "",
      caption = custom_caption,
      col.names = c("Effect", "Estimate", "Lower 95\\% CI", "Upper 95\\% CI",
                    "Std. Error", "$df$", "$t$", "$p$"),
      escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
footnote(general = paste0("$R^2_{conditional}$ = ",
                          round(r2_nakagawa(mod)$R2_conditional, 3),
                          ", $R^2_{marginal}$ = ",
                          round(r2_nakagawa(mod)$R2_marginal, 3),
                          ". Results are from linear mixed models for main
                          effects and interactions between sexual desire (SD) dimensions,
                          sex, and Stimuli sex.
                          Confidence intervals were calculated as the 2.5 and 97.5
                          percentiles from bootstrap (1000 simulations).
                          Continuous variables were centered and scaled
                          (represented as \\textbf{(C)} in variable names).
                          Gender = participants gender (women, men);
                          Stimuli sex = sex of stimuli (female, male);
                          Solitary SD = Solitary Sexual Desire;
                          Attractive person DSD = Dyadic Sexual Desire toward an
                          Attractive person;
                          Partner DSD = Dyadic Sexual Desire toward partner.
                          \\textit{Sum-to-zero} contrasts were used to display
                          \\textit{p}-values that represent main effects and interactions
                          in an ANOVA-type manner (i.e. the intercept is the grand mean of
                          all cells, and estimates are differences between each category
                          mean and the mean of all categories).
                          As reference categories
                          \\textit{Single} was used for relationship status,
                          \\textit{Men} for gender,
                          and \\textit{Male} for stimuli sex.
                          Contrasted levels are in square brackets.
                          Significant effects are in bold."),
          escape = FALSE,
          threeparttable = TRUE,
          footnote_as_chunk = TRUE)
return(modTab)
}

```

1.3.5 emms.sig

Function to create a table of estimated marginal means and contrasts at three levels of a covariate, representing significance levels from `emmeans::emmeans` outputs. The function highlights significant p values, and formats the output in L^AT_EX, ready to be used with `kable`.

```

# Version 1, for interactions
emms.sig <- function(low.i, mid.i, hi.i) {
  emm.low <- data.frame(low.i[[1]])
  emm.mid <- data.frame(mid.i[[1]])
  emm.hi <- data.frame(hi.i[[1]])
  con.low <- data.frame(low.i[[2]])
  con.mid <- data.frame(mid.i[[2]])
  con.hi <- data.frame(hi.i[[2]])
}

```

```

low.tab <- merge(emm.low, con.low, by = 0, all = TRUE)
mid.tab <- merge(emm.mid, con.mid, by = 0, all = TRUE)
hi.tab <- merge(emm.hi, con.hi, by = 0, all = TRUE)

tab <- bind_rows(low.tab, mid.tab, hi.tab) |>
  select(-c(1,3,6,10:13)) |>
  mutate(p.value = pval.lev(p.value)) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 4), "l", rep("c", 2)),
        linesep = "",
        caption = paste0("Estimated marginal means and contrasts for ",
                          low.i[[1]]@misc$pri.vars[1],
                          " at different levels of ",
                          low.i[[1]]@misc$by.vars),
        col.names = c(low.i[[1]]@misc$pri.vars[1],
                      "EMM", "$SE$", "$2.5\\% CI$", "$97.5\\% CI$", "Contrast", "$z$", "$p$"),
        escape = FALSE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean - SD"),
            start_row = 1,
            end_row = 2,
            bold = TRUE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean"),
            start_row = 3,
            end_row = 4,
            hline_before = TRUE,
            bold = TRUE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean + SD"),
            start_row = 5,
            end_row = 6,
            hline_before = TRUE,
            bold = TRUE) |>
  add_header_above(c(" " = 5, "Contrasts" = 3)) |>
  kable_styling(latex_options = "HOLD_position") |>
  footnote(general = paste0("EMM = estimated marginal mean.
    Significant effects are in bold.
    Continuous variables were centered and scaled (in this case, ",
    low.i[[1]]@misc$by.vars, ").
    An asymptotic method was used to avoid extreme computation
    times (hence, no degrees of freedom are included, and
    $z$ rather than $t$ statistics are reported).
    For contrasts, Tukey adjustment was used."),
    threeparttable = TRUE,
    footnote_as_chunk = TRUE,
    escape = FALSE)

  return(tab)
}

# Version 2, for triple interactions
emms.sig2 <- function(low.i, mid.i, hi.i) {
  emm.low <- data.frame(low.i[[1]])
  emm.mid <- data.frame(mid.i[[1]])
  emm.hi <- data.frame(hi.i[[1]])
  con.low <- data.frame(low.i[[2]])

```

```

con.mid <- data.frame(mid.i[[2]])
con.hi <- data.frame(hi.i[[2]])

low.tab <- merge(emm.low, con.low, by = 0, all = TRUE)
mid.tab <- merge(emm.mid, con.mid, by = 0, all = TRUE)
hi.tab <- merge(emm.hi, con.hi, by = 0, all = TRUE)

tab <- bind_rows(low.tab, mid.tab, hi.tab) |>
  select(-c(1,4,7,11:14)) |>
  mutate(p.value = pval.lev(p.value)) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", "l", rep("c", 4), "l", rep("c", 2)),
        linesep = "",
        caption = paste0("Estimated marginal means and contrasts for ",
                          low.i[[1]]@misc$pri.vars[1], " and ",
                          low.i[[1]]@misc$pri.vars[2],
                          " at different levels of ",
                          low.i[[1]]@misc$by.vars),
        col.names = c(low.i[[1]]@misc$pri.vars[1],
                      low.i[[1]]@misc$pri.vars[2],
                      "EMM", "$SE$", "$2.5\\% CI$", "$97.5\\% CI$", "Contrast", "$z$", "$p$"),
        escape = FALSE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean - SD"),
            start_row = 1,
            end_row = 6,
            bold = TRUE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean"),
            start_row = 7,
            end_row = 12,
            hline_before = TRUE,
            bold = TRUE) |>
  pack_rows(group_label = paste0(low.i[[1]]@misc$by.vars, " = Mean + SD"),
            start_row = 13,
            end_row = 18,
            hline_before = TRUE,
            bold = TRUE) |>
  add_header_above(c(" " = 6, "Contrasts" = 3)) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  footnote(general = paste0("EMM = estimated marginal mean.
    Significant effects are in bold.
    Continuous variables were centered and scaled (in this case, ",
    low.i[[1]]@misc$by.vars, ")
    An asymptotic method was used to avoid extreme computation
    times (hence, no degrees of freedom are included, and
    $z$ rather than $t$ statistics are reported).
    For contrasts, Tukey adjustment was used."),
    threeparttable = TRUE,
    footnote_as_chunk = TRUE,
    escape = FALSE)

return(tab)
}

```


1.3.6 contr.stars

Function to create a data frame of model contrasts, representing significance levels from an `emmeans::emmeans` output. These data frames are formatted to be called by the `ggpubr::stat_pvalue_manual` function used in model figures.

```
contr.stars <- function(emms){
  require(emmeans)
  x <- as.data.frame(contrast(emms, interaction = "pairwise"))
  x <- separate(x,
    col = 1,
    into = c("group1", "group2"),
    sep = " - ",
    remove = TRUE)
  x$p.signif <- ifelse(x$p.value < 0.0001, "****",
    ifelse(x$p.value < 0.001, "***",
      ifelse(x$p.value < 0.01, "**",
        ifelse(x$p.value < 0.05, "*", NA))))
  x <- x |>
    mutate_at("group1", str_replace_all, "[()]", "") |>
    mutate_at("group2", str_replace_all, "[()]", "")
  return(x)
}
```

1.3.7 prob.dist.tab

Function to create a table of the probability of a model for each distribution family, using the `check_distribution` function, from the `performance` package (Lüdtke et al., 2021). Values are sorted descending, first for probabilities according to the residual distribution, and then for probabilities according to the response variable. While 18 distribution families are tested, only families with at least one probability (either residual or response variable) higher than 10% are shown in the table.

```
prob.dist.tab <- function(mod){
  # Calculate probabilities for each distribution family
  tibble(check_distribution(mod)) |>
    arrange(desc(p_Response)) |>
    arrange(desc(p_Residuals)) |>
  # Select only distribution families with at least a 10% probability
  filter(p_Residuals > 0.1 | p_Response > 0.1) |>
  # Transform probabilities to percentages
  mutate(p_Residuals = paste0(round(p_Residuals*100, 2), "\\%")) |>
  mutate(p_Response = paste0(round(p_Response*100, 2), "\\%")) |>
  # Capitalise first letter of each family distribution
  mutate(Distribution = sub("(.)", "\\U\\1", Distribution, perl = TRUE)) |>
  # Create table
  kable(booktabs = TRUE,
    align = c("l", "c", "c"),
    row.names = FALSE,
    caption = "Distributional family for the model",
    col.names = c("Family",
      "Residuals",
      "Response"),
    escape = FALSE) |>
  kable_styling(latex_options = "HOLD_position") |>
  # Bold highest probability
  row_spec(1, background = "#c4c4c4") |>
  footnote(general = "Only families with at least one probability higher than
  10\\% are shown, but a total of 18 distribution families were tested.
```

```

    The most likely distribution is highlighted.",
    threeparttable = TRUE,
    footnote_as_chunk = TRUE,
    escape = FALSE)
}

```

1.4 Load and wrangle data

Change necessary variables to factor, sort levels, and rename variables

```

# Load data
dat <- read.csv("Data/BD_Heterosexuales_Vertical_BIG.csv") |>
# Remove rows with missing values for Solitary sexual desire (SD_solitario)
drop_na(SD_solitario) |>
# Change variables to factor and sort their levels
mutate_at(c("Contenido_Estimulo", "Sexo", "Sexo_Estimulo", "PrefSex", "EstRel", "Escolaridad",
            "Religion", "TiempoRP"), as.factor) |>
# Rename variables to English
rename(Participant = Participante,
       Age = EdadParticipante,
       `Preferred sex` = PrefSex,
       Gender = Sexo,
       `Contraceptive uso` = Anticoncep,
       `Last period` = UltimoPer,
       `Period day` = Dia_ciclo,
       Education = Escolaridad,
       Location = Residencia,
       `Location (other)` = Residencia_3_TEXT,
       `Medical history` = AntMed,
       `Sexual orientation` = OS,
       `Relationship status` = EstRel,
       `Relationship duration` = TiempoRP,
       `Partner gender` = SexPareja,
       `Relationship type` = TipoRel,
       `Age at first intercourse` = Primera.ExpSex,
       `Consented to first intercourse` = ConExpSex,
       `Number of sexual partners` = Numero.Parejas,
       `Pornography consumed last month` = Pornografia_ultimo_mes,
       Relationship = TieneRelacion,
       `MGH-SFQ (total)` = MGH.SFQ_Total,
       `Dyadic sexual desire (Partner)` = SD_Diadico_pareja,
       `Solitary sexual desire` = SD_solitario,
       `Dyadic sexual desire (Attractive person)` = SD_Diadico_p_atractiva,
       `MGSS sexual satisfaction (General)` = Satisfaccion.Sexual..MGSS_general.,
       `MGSS sexual satisfaction (Partner)` = Satisfaccion.Sexual..MGSS_Pareja.,
       `Stimuli code` =Codigo_Estimulo,
       `Stimuli sex` = Sexo_Estimulo,
       `Stimuli content` = Contenido_Estimulo,
       `Subjective sexual attractiveness` = Atractivo,
       `Subjective sexual arousal` = Excitacion) |>
# Recode factor levels
mutate(`Stimuli content` = recode_factor(`Stimuli content`,
                                         Erotico = "Erotic",
                                         No_erotico = "Non-erotic")) |>
mutate(Gender = recode_factor(Gender,
                              Femenino = "Women",
                              Masculino = "Men")) |>

```

```

mutate(`Stimuli sex` = recode_factor(`Stimuli sex`,
                                   Femenino = "Female",
                                   Masculino = "Male")) |>
mutate(`Preferred sex` = recode_factor(`Preferred sex`,
                                       Hombre = "Male",
                                       Mujer = "Female")) |>
mutate(Education = recode(Education,
                          "Bachillerato" = "High school",
                          "Universitario" = "University",
                          "Postgrado" = "Postgraduate")) |>
mutate(Religion = recode(Religion,
                         "1" = "Religious",
                         "0" = "Non-religious")) |>
mutate(`Pornography consumed last month` = recode(`Pornography consumed last month`,
                                                  "Nunca" = "None",
                                                  "Una o dos veces" = "1-2 times",
                                                  "Tres a cinco veces" = "3-5 times",
                                                  "Mas de 5 veces" = "5 times or more")) |>

# Recode relationship duration
#mutate(`Relationship duration` = replace_na(`Relationship duration`, "Single"))
mutate(`Relationship duration` = recode(`Relationship duration`,
                                       "Sin pareja actual" = "Single",
                                       "Menor a 6 meses" = "Less that 6 months",
                                       "Entre 6 meses y 2 anos" = "Between 6 months and 2 years",
                                       "Entre 2 y 5 anos" = "Between 2 and 5 years",
                                       "Más de 5 anos" = "More than 5 years"),
      `Relationship duration` = replace_na(`Relationship duration`, "Single")) |>
# Recode relationship type
mutate(Relationship = recode(`Relationship status`,
                             "Exclusiva/No viven juntos" = "Stable",
                             "Exclusiva/Matrimonio" = "Stable",
                             "No exclusiva" = "Non-stable",
                             "Soltero/sin contactos sexuales en un ano" = "Single",
                             "Soltero/contactos sexuales en un ano" = "Single")) |>

# Relevel factors
mutate(Education = fct_relevel(Education,
                              c("High school", "University", "Postgraduate")),
      `Pornography consumed last month` = fct_relevel(`Pornography consumed last month`,
                                                    c("None", "1-2 times",
                                                      "3-5 times", "5 times or more")),
      `Relationship duration` = fct_relevel(`Relationship duration`,
                                           c("Single", "Less that 6 months",
                                             "Between 6 months and 2 years",
                                             "Between 2 and 5 years",
                                             "More than 5 years")) |>
mutate(`Stimuli content` = as.factor(`Stimuli content`),
      `Stimuli sex` = as.factor(`Stimuli sex`)) |>
# Filter participants in non-stable relationships
filter(Relationship != "Non-stable") |>
droplevels()

```

2 Descriptives

2.0.1 Figure S1. Demographic characteristics of the sample

Number of participants by demographic category.

```
# Get number of participant for each combination of demographic characteristic
dat.demog <- dat |>
  select(Participant, Gender, Relationship, Education, Religion,
         `Pornography consumed last month`) |>
  group_by(Participant) |>
  filter(row_number() == 1) |>
  ungroup() |>
  group_by(Gender, Relationship, Education, Religion,
         `Pornography consumed last month`) |>
  rename(Porn = `Pornography consumed last month`) |>
  tally() |>
  drop_na(Religion) |>
  ungroup()

# Create separate tables by gender
dat.demog.W <- filter(dat.demog, Gender == "Women")
dat.demog.M <- filter(dat.demog, Gender == "Men")

# Women
samp.w <- ggballoonplot(dat.demog.W, x = "Education", y = "Porn", size = "n",
  fill = "n",
  facet.by = c("Relationship", "Religion")) +
  scale_fill_viridis_c(option = "C", limits = c(1, max(dat.demog$n))) +
  scale_size_continuous(range = c(1, 7), limits = c(1, max(dat.demog$n))) +
  guides(fill = guide_legend(face = "italic"),
         size = guide_legend(face = "italic")) +
  labs(title = "Women", y = "Pornography consumed last month") +
  geom_text(aes(label = n),
    size = 3, nudge_x = 0.3, nudge_y = 0.1) +
  geom_text(aes(label = paste0("\n(",
                                percent(n/sum(dat.demog$n), accuracy = 0.1),
                                ")")),
    size = 2.5, nudge_x = 0.3, nudge_y = -0.05) +
  theme_tq() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(angle = 45, vjust = 0.5))

# Men
samp.m <- ggballoonplot(dat.demog.M, x = "Education", y = "Porn", size = "n",
  fill = "n",
  facet.by = c("Relationship", "Religion")) +
  scale_fill_viridis_c(option = "C", limits = c(1, max(dat.demog$n))) +
  scale_size_continuous(range = c(1, 7), limits = c(1, max(dat.demog$n))) +
  guides(fill = guide_legend(face = "italic"),
         size = guide_legend(face = "italic")) +
  labs(title = "Men", y = NULL) +
  geom_text(aes(label = n),
    size = 3, nudge_x = 0.3, nudge_y = 0.1) +
  geom_text(aes(label = paste0("\n(",
                                percent(n/sum(dat.demog$n), accuracy = 0.1),
                                ")")),
    size = 2.5, nudge_x = 0.3, nudge_y = -0.05) +
```

```

theme_tq() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(angle = 45, vjust = 0.5))

# Full plot
ggarrange(samp.w, samp.m,
  widths = c(1.1, 1),
  common.legend = TRUE,
  legend = "bottom")

```



Figure S1. Number of participants by gender (left = women, right = men), Relationship (stable = top panels, single = bottom panels), Religion (non-religious = left panels by gender, religious = right panels by gender), Education (X axis), and pornography consumed during the last month (Y axis). The number of participants for each combination of these five variables is displayed as numbers (percentage in brackets), as well as by the color and size of the bubbles.

2.1 Descriptive statistics of the participants by gender

Calculate mean values per participant for relevant, numeric variables.

```

# Summarize relevant variables by participant
dat.desc <- dat |>
  select(Participant, Gender, Age, Relationship, `Number of sexual partners`,
    `MGH-SFQ (total)`,
    `MGSS sexual satisfaction (General)`, `MGSS sexual satisfaction (Partner)`,
    `Subjective sexual attractiveness`, `Subjective sexual arousal`,
    `Solitary sexual desire`,
    `Dyadic sexual desire (Attractive person)`, `Dyadic sexual desire (Partner)`) |>
  group_by(Participant, Gender, Relationship) |>
  summarize_if(is.numeric, mean, na.rm = TRUE)

```

2.1.1 Table S1. Descriptive statistics of the participants by gender

Table of descriptives by gender.

```
# Table of descriptives by gender and relationship status
describeBy(dat.desc ~ Relationship + Gender,
           mat=TRUE,
           digits=2) |>
rownames_to_column("Measured characteristic") |>
select(1,3:4,6:9,12:13) |>
slice(-(1:12)) |>
select(1,3,2,4:9) |>
# Remove numbers included to differentiate repeated row names (now on column 1)
mutate("Measured characteristic" = str_replace_all(`Measured characteristic`,
                                                    c("1" = "", "2" = "", "3" = "", "4" = ""))) |>

# Create table
kable(digits = 2,
      booktabs = TRUE,
      align = c("l", "l", rep("c", 7)),
      linesep = "",
      caption = "Descriptive statistics the participants by gender
and relationship status",
      col.names = c("Measured characteristic", "Gender", "Relationship status",
                    "$n$", "Mean", "$SD$", "Median", "Min", "Max"),
      longtable = TRUE,
      escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position"),
              font_size = 8.2) |>
collapse_rows(columns = 1:3, valign = "middle") |>
footnote(general = "Because for \\\textit{Subjective sexual attractiveness} and
\\\textit{Subjective sexual arousal} there are are multiple within-subject
observations, descriptives are calculated from mean values per participant.",
         threeparttable = TRUE,
         footnote_as_chunk = TRUE,
         escape = FALSE)
```

Table S1. *Descriptive statistics the participants by gender and relationship status*

Measured characteristic	Gender	Relationship status	<i>n</i>	Mean	<i>SD</i>	Median	Min	Max
Age	Women	Stable	105	24.51	5.58	23.00	18.00	40.00
		Single	79	22.27	3.84	21.00	18.00	36.00
	Men	Stable	72	26.72	5.64	25.00	19.00	40.00
		Single	67	24.24	4.58	23.00	18.00	39.00
Number of sexual partners	Women	Stable	103	4.41	3.77	3.00	1.00	22.00
		Single	76	5.74	8.85	3.00	0.00	63.00
	Men	Stable	72	8.72	11.36	5.00	1.00	70.00
		Single	66	7.30	8.06	4.00	0.00	40.00
MGH-SFQ (total)	Women	Stable	104	3.31	0.96	3.75	0.00	4.00
		Single	79	2.80	1.23	3.50	0.00	4.00
	Men	Stable	72	3.59	0.62	3.90	0.60	4.00
		Single	67	3.38	0.83	3.80	0.60	4.00
	Women	Stable	100	25.88	5.67	28.00	6.00	30.00
		Single	10	26.90	3.11	27.00	22.00	30.00
		Stable	70	26.43	4.54	29.00	12.00	30.00

MGSS sexual satisfaction (General)								
MGSS sexual satisfaction (Partner)	Men	Single	12	23.58	5.14	24.50	14.00	29.00
		Stable	100	28.13	4.20	30.00	8.00	30.00
	Women	Single	10	28.10	2.13	29.00	25.00	30.00
		Stable	70	28.49	3.48	30.00	6.00	30.00
Subjective sexual attractiveness	Men	Single	12	26.08	4.85	27.50	15.00	30.00
		Stable	105	2.94	1.11	2.78	1.00	5.49
	Women	Single	79	3.19	1.06	3.11	1.44	6.77
		Stable	72	3.27	0.94	3.24	1.11	6.20
Subjective sexual arousal	Men	Single	67	3.20	0.90	3.18	1.09	5.72
		Stable	105	1.59	0.68	1.39	1.00	4.21
	Women	Single	79	1.75	0.71	1.52	1.00	4.39
		Stable	72	2.24	0.83	2.07	1.00	4.57
Solitary sexual desire	Men	Single	67	2.16	0.78	2.05	1.00	4.09
		Stable	105	11.53	8.59	12.00	0.00	29.00
	Women	Single	79	16.03	8.35	17.00	0.00	31.00
		Stable	72	17.47	7.51	17.50	0.00	31.00
Dyadic sexual desire (Attractive person)	Men	Single	67	18.25	7.10	19.00	1.00	31.00
		Stable	105	10.55	7.64	10.00	0.00	30.00
	Women	Single	79	14.06	7.39	15.00	0.00	32.00
		Stable	72	16.21	7.44	15.50	0.00	32.00
Dyadic sexual desire (Partner)	Men	Single	67	17.57	6.66	17.00	2.00	30.00
		Stable	105	27.53	8.50	30.00	0.00	38.00
	Women	Single	76	21.33	10.91	23.00	0.00	38.00
		Stable	72	31.35	5.33	32.00	15.00	38.00
		Single	67	25.81	9.40	28.00	0.00	38.00

Note: Because for *Subjective sexual attractiveness* and *Subjective sexual arousal* there are multiple within-subject observations, descriptives are calculated from mean values per participant.

2.1.2 Figure S2. Distribution of participants' measured variables by gender

Kernel density distributions by gender.

```
# Convert dat.desc to long format
datp <- dat.desc |>
  pivot_longer(cols = Age:`Dyadic sexual desire (Partner)`,
    names_to = "Variable",
    values_to = "Value") |>
  mutate(Variable = str_wrap(Variable, width = 30))

# Figure created as 3 separate panels (to use a different number of panels per row)
fs2a <- ggplot(datp |>
  filter(Variable %in% c("Age",
    "Number of sexual partners",
    "Subjective sexual\nattractiveness",
    "Subjective sexual arousal")),
  aes(Value,
    fill = Gender,
    colour = Gender)) +
  geom_density(alpha = 0.3) +
  geom_vline(data = datp |>
    filter(Variable %in% c("Age",
```

```

                                "Number of sexual partners",
                                "Subjective sexual\nattractiveness",
                                "Subjective sexual arousal")) |>
  group_by(Variable, Gender) |>
  summarise(mean = mean(Value, na.rm =TRUE)),
  size = 1,
  aes(xintercept = mean, color = Gender, linetype = Gender)) +
scale_color_manual(values = color.Gender) +
scale_fill_manual(values = color.Gender) +
facet_wrap(~ Variable,
  scales = "free",
  ncol = 4) +
labs(y = "Density",
  x = NULL) +
theme_tq()

fs2b <- ggplot(datp |>
  filter(Variable %in% c("MGH-SFQ (total)",
    "MGSS sexual satisfaction\n(General)",
    "MGSS sexual satisfaction\n(Partner)")),
  aes(Value,
    fill = Gender,
    colour = Gender)) +
geom_density(alpha = 0.3) +
geom_vline(data = datp |>
  filter(Variable %in% c("MGH-SFQ (total)",
    "MGSS sexual satisfaction\n(General)",
    "MGSS sexual satisfaction\n(Partner)")), |>
  group_by(Variable, Gender) |>
  summarise(mean = mean(Value, na.rm =TRUE)),
  size = 1,
  aes(xintercept = mean, color = Gender, linetype = Gender)) +
scale_color_manual(values = color.Gender) +
scale_fill_manual(values = color.Gender) +
facet_wrap(~ Variable,
  scales = "free",
  ncol = 3) +
labs(y = "Density",
  x = NULL) +
theme_tq()

fs2c <- ggplot(datp |>
  filter(Variable %in% c("Solitary sexual desire",
    "Dyadic sexual desire\n(Attractive person)",
    "Dyadic sexual desire (Partner)")),
  aes(Value,
    fill = Gender,
    colour = Gender)) +
geom_density(alpha = 0.3) +
geom_vline(data = datp |>
  filter(Variable %in% c("Solitary sexual desire",
    "Dyadic sexual desire\n(Attractive person)",
    "Dyadic sexual desire (Partner)")), |>
  group_by(Variable, Gender) |>
  summarise(mean = mean(Value, na.rm =TRUE)),
  size = 1,

```



```

aes(xintercept = mean, color = Gender, linetype = Gender)) +
scale_color_manual(values = color.Gender) +
scale_fill_manual(values = color.Gender) +
facet_wrap(~ Variable,
            scales = "free",
            ncol = 3) +
labs(y = "Density",
     x = NULL) +
theme_tq()

# Full plot
ggarrange(fs2a, fs2b, fs2c,
          common.legend = TRUE,
          legend = "bottom",
          nrow = 3)

```

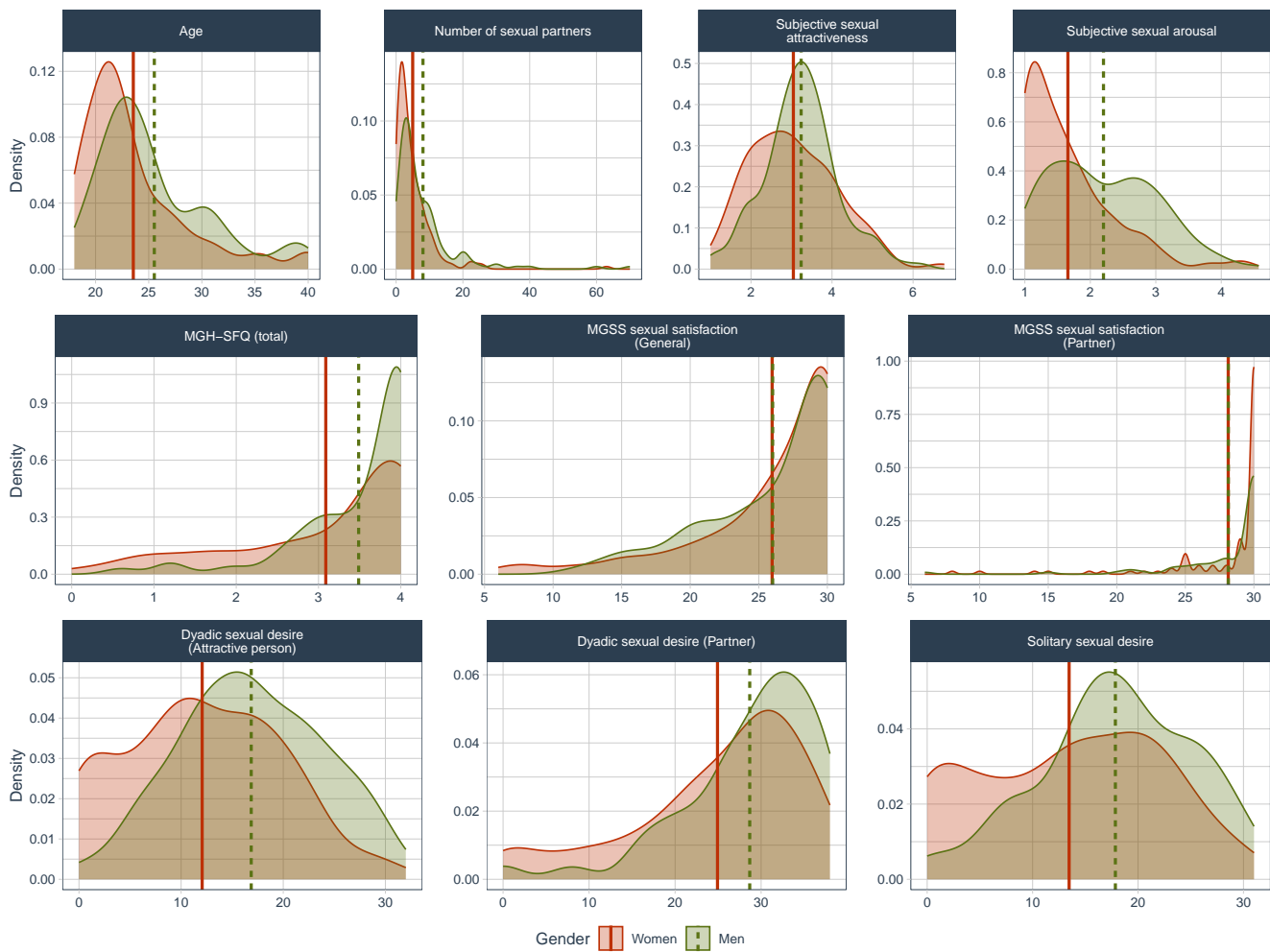


Figure S2. Distribution of measured variables by gender. Coloured vertical lines represent mean values by gender. Detailed descriptives are found in Table S1. Because for *Subjective sexual attractiveness* and *Subjective sexual arousal* there are multiple within-subject observations, densities calculated from mean values per participant.

2.2 Correlations between measured variables

Correlation between numeric variables for women, men, and all participants combined, are reported in Table S2.

2.2.1 Table S2. Correlations between measured variables

Correlation matrix table.

```
# Correlations for women
dat.corr.W <- dat.desc |>
  ungroup() |>
  filter(Gender == "Women") |>
  select(Age:`Dyadic sexual desire (Partner)` ) |>
  corr.stars() |>
  rownames_to_column(var = " ")

# Correlations for men
dat.corr.M <- dat.desc |>
  ungroup() |>
  filter(Gender == "Men") |>
  select(Age:`Dyadic sexual desire (Partner)` ) |>
  corr.stars() |>
  rownames_to_column(var = " ")

# Correlations for all participants combined
dat.corr.All <- dat.desc |>
  ungroup() |>
  select(Age:`Dyadic sexual desire (Partner)` ) |>
  corr.stars() |>
  rownames_to_column(var = " ")

# Full formatted table
bind_rows(dat.corr.W, dat.corr.M, dat.corr.All) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 9)),
        linesep = "",
        caption = "Correlations between measured variables",
        escape = FALSE) |>
  pack_rows(group_label = "Women",
            start_row = 1, end_row = 10,
            bold = TRUE) |>
  pack_rows(group_label = "Men",
            start_row = 11, end_row = 20,
            hline_before = TRUE,
            bold = TRUE) |>
  pack_rows(group_label = "All participants",
            start_row = 21, end_row = 30,
            hline_before = TRUE,
            bold = TRUE) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  column_spec(2:10, width = "2.2cm") |>
  footnote(general = paste0("Values represent Pearson correlation coefficients ($r$). ",
                            "For significance, $^{\dagger}p$ < 0.1, *$p$ < 0.05, ",
                            "**$p$ < 0.01, ***$p$ < 0.001. ",
                            "Significant correlations are in bold."),
          threeparttable = TRUE,
          footnote_as_chunk = TRUE,
          escape = FALSE) |>
  landscape()
```

Table S2. Correlations between measured variables

	Age	Number of sexual partners	MGH-SFQ (total)	MGSS sexual satisfaction (General)	MGSS sexual satisfaction (Partner)	Subjective sexual attractiveness	Subjective sexual arousal	Solitary sexual desire	Dyadic sexual desire (Attractive person)
Women									
Age									
Number of sexual partners	0.24**								
MGH-SFQ (total)	-0.05	-0.07							
MGSS sexual satisfaction (General)	-0.21*	0.02	0.46***						
MGSS sexual satisfaction (Partner)	-0.16 [†]	-0.14	0.32***	0.73***					
Subjective sexual attractiveness	0.11	0.18*	-0.04	-0.22*	-0.18 [†]				
Subjective sexual arousal	0.00	0.17*	-0.13 [†]	-0.18 [†]	-0.16 [†]	0.54***			
Solitary sexual desire	-0.14 [†]	0.28***	0.05	-0.06	-0.18 [†]	0.31***	0.33***		
Dyadic sexual desire (Attractive person)	0.06	0.32***	-0.17*	-0.04	-0.17 [†]	0.34***	0.36***	0.44***	
Dyadic sexual desire (Partner)	0.00	0.21**	0.43***	0.44***	0.27**	0.13 [†]	0.04	0.31***	0.13 [†]
Men									
Age									
Number of sexual partners	0.23**								
MGH-SFQ (total)	0.04	0.02							
MGSS sexual satisfaction (General)	-0.24*	-0.08	0.36***						
MGSS sexual satisfaction (Partner)	-0.13	-0.01	0.10	0.63***					
Subjective sexual attractiveness	0.10	-0.05	-0.08	-0.10	-0.02				
Subjective sexual arousal	0.2*	0.07	0.05	-0.14	-0.09	0.46***			
Solitary sexual desire	-0.16 [†]	0.00	0.09	0.10	0.17	0.26**	0.11		
Dyadic sexual desire (Attractive person)	0.12	0.29***	0.03	-0.13	-0.08	0.25**	0.43***	0.25**	
Dyadic sexual desire (Partner)	0.11	0.07	0.36***	0.55***	0.22*	0.14	0.24**	0.17*	0.2*
All participants									
Age									
Number of sexual partners	0.26***								
MGH-SFQ (total)	0.02	0.01							
MGSS sexual satisfaction (General)	-0.22**	-0.03	0.42***						
MGSS sexual satisfaction (Partner)	-0.14*	-0.07	0.24***	0.69***					
Subjective sexual attractiveness	0.12*	0.08	-0.03	-0.18*	-0.12				
Subjective sexual arousal	0.15**	0.17**	0.01	-0.15*	-0.12 [†]	0.5***			
Solitary sexual desire	-0.09	0.17**	0.11 [†]	0.00	-0.05	0.31***	0.3***		
Dyadic sexual desire (Attractive person)	0.14*	0.33***	-0.04	-0.07	-0.12 [†]	0.32***	0.45***	0.42***	
Dyadic sexual desire (Partner)	0.08	0.16**	0.43***	0.46***	0.25***	0.15**	0.18**	0.3***	0.21***

Note: Values represent Pearson correlation coefficients (r). For significance, [†] $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Significant correlations are in bold.

2.3 Internal consistency

Six variables were calculated from multiple items (1. MGH-SFQ, 2. Dyadic sexual desire (Partner), 3. Solitary sexual desire, 4. Dyadic sexual desire (Attractive person), 5. MGSS sexual satisfaction (General) and 6. MGSS sexual satisfaction (Partner)).

Data by item, for each participant, is included in the following data base, loaded as `dat.reli`:

```
dat.reli <- read_excel("Data/BD_ConsistenciaInterna.xlsx") |>
  mutate(Sex = recode_factor(Sex,
                             "2" = "Women",
                             "1" = "Men")) |>
  rename(Gender = Sex) |>
  filter(Participante != 122)
```

Participant 122 was excluded because they did not respond the psychological scales.

To measure the internal consistency of these tests, we used standardized Cronbach's alpha (α or Tau-equivalent reliability: ρ_T) coefficients, using the function `cronbach.alpha` from the package `ltm` (Rizopoulos, 2006).

Importantly, given that for MGH-SFQ one item was answered only by men, the internal consistency of this variable was measured independently for each gender.

```
# MGH-SFQ for men
MGH.m <- dat.reli |>
  filter(Gender == "Men" ) |>
  select(3:7) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# MGH-SFQ for women
MGH.w <- dat.reli |>
  filter(Gender == "Women" ) |>
  select(3:5,7) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Dyadic sexual desire (Partner)
DSD.p <- dat.reli |>
  select(9:13) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Solitary sexual desire
SSD.p <- dat.reli |>
  select(15:18) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Dyadic sexual desire (Attractive person)
DSD.a <- dat.reli |>
  select(20:23) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# MGSS sexual satisfaction (General)
MGSS.g <- dat.reli |>
  select(26:30) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)
```

```
# MGSS sexual satisfaction (Partner)
MGSS.p <- dat.reli |>
  select(32:36) |>
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)
```

2.3.1 Table S3. Internal consistency of construct variables

Table of Cronbach's α for construct variables.

```
# Create table
tibble(Variable = c("MGH-SFQ", "MGH-SFQ",
                    "MGSS sexual satisfaction (General)",
                    "MGSS sexual satisfaction (Partner)",
                    "Dyadic sexual desire (Partner)",
                    "Solitary sexual desire",
                    "Dyadic sexual desire (Attractive person)"),
       Gender = c("Men", "Women", rep(" ", 5)),
       p = c(MGH.m$p,
             MGH.w$p,
             MGSS.g$p,
             MGSS.p$p,
             DSD.p$p,
             SSD.p$p,
             DSD.a$p),
       n = c(MGH.m$n,
             MGH.w$n,
             MGSS.g$n,
             MGSS.p$n,
             DSD.p$n,
             SSD.p$n,
             DSD.a$n),
       alpha = c(MGH.m$alpha,
                 MGH.w$alpha,
                 MGSS.g$alpha,
                 MGSS.p$alpha,
                 DSD.p$alpha,
                 SSD.p$alpha,
                 DSD.a$alpha),
       ci2.5 = c(MGH.m$ci[1],
                 MGH.w$ci[1],
                 MGSS.g$ci[1],
                 MGSS.p$ci[1],
                 DSD.p$ci[1],
                 SSD.p$ci[1],
                 DSD.a$ci[1]),
       ci97.5 = c(MGH.m$ci[2],
                  MGH.w$ci[2],
                  MGSS.g$ci[2],
                  MGSS.p$ci[2],
                  DSD.p$ci[2],
                  SSD.p$ci[2],
                  DSD.a$ci[2])) |>
kable(digits = 2,
      booktabs = TRUE,
      align = c("l", "l", rep("c", 5)),
```

```

linesep = "",
caption = "Internal consistency of measured variables",
escape = FALSE,
col.names = c("Variable", "Gender",
              "Items",
              "$n$",
              "$\\alpha$",
              "$2.5\\% CI$",
              "$97.5\\% CI$")) |>
collapse_rows(columns = 1, valign = "middle") |>
kable_styling(latex_options = "HOLD_position") |>
footnote(general = "95\\% confidence intervals were calculated with 1,000 bootstrap samples.
Standardized Cronbach's alpha ($\\alpha$) coefficients were computed.
MGH-SFQ is reported by gender, because one item was answered only by men.",
threeparttable = TRUE,
footnote_as_chunk = TRUE,
escape = FALSE)

```

Table S3. Internal consistency of measured variables

Variable	Gender	Items	<i>n</i>	α	2.5%CI	97.5%CI
MGH-SFQ	Men	5	139	0.82	0.71	0.89
	Women	4	181	0.86	0.82	0.90
MGSS sexual satisfaction (General)		5	188	0.92	0.88	0.94
MGSS sexual satisfaction (Partner)		5	187	0.91	0.85	0.95
Dyadic sexual desire (Partner)		5	309	0.90	0.87	0.92
Solitary sexual desire		4	314	0.91	0.89	0.93
Dyadic sexual desire (Attractive person)		4	320	0.89	0.87	0.91

Note: 95% confidence intervals were calculated with 1,000 bootstrap samples. Standardized Cronbach's alpha (α) coefficients were computed. MGH-SFQ is reported by gender, because one item was answered only by men.

3 Controlling the effects of Relationship Duration and MGSS sexual satisfaction (Partner) on sexual desire dimensions

We standardized the scores of all three sexual desire dimensions so that they were not correlated with Relationship Duration or MGSS sexual satisfaction (Partner). To do this, we performed two steps.

First, we conducted linear regression analyses to identify the relationships between Relationship Duration and MGSS sexual satisfaction (Partner) on each of the three sexual desire dimensions. Next, we made the effect zero, but kept the model residuals, to control for the effects of Relationship Duration and MGSS sexual satisfaction (Partner). By doing so, we effectively removed their influence on the resulting scores, leaving us with adjusted versions of the three desire dimensions that are not correlated with these external variables. MGSS sexual satisfaction (Partner) was mean centered.

```

# Create a subset of the data that only includes stable relationships
dat_ctl <- dat |>
  # Group the data by participant and then select one (the top) observation for each one
  group_by(Participant) |>
  slice_head() |>
  filter(Relationship == "Stable") |>

```

```

ungroup()

# Fit a linear model to predict solitary sexual desire based on relationship duration and partner satisfaction
ctl_SSD <- lm(`Solitary sexual desire` ~
              `Relationship duration` + `MGSS sexual satisfaction (Partner)`,
              data = dat_ctl)

# Perform an ANOVA test for the CTL SSD model
Anova(ctl_SSD, type = 3)

## Anova Table (Type III tests)
##
## Response: Solitary sexual desire
##
##              Sum Sq Df F value    Pr(>F)
## (Intercept)      614.3   1  8.1789 0.004786 **
## 'Relationship duration'      108.6   3  0.4822 0.695135
## 'MGSS sexual satisfaction (Partner)'      2.2   1  0.0289 0.865254
## Residuals      12393.5 165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Fit a model to predict dyadic sexual desire based on relationship duration and partner satisfaction
ctl_PD <- lm(`Dyadic sexual desire (Partner)` ~
            `Relationship duration` + `MGSS sexual satisfaction (Partner)`,
            data = dat_ctl)

# Perform an ANOVA test for the CTL PD model
Anova(ctl_PD, type = 3)

## Anova Table (Type III tests)
##
## Response: Dyadic sexual desire (Partner)
##
##              Sum Sq Df F value    Pr(>F)
## (Intercept)      844.3   1 18.3715 3.082e-05 ***
## 'Relationship duration'      286.9   3  2.0807  0.104744
## 'MGSS sexual satisfaction (Partner)'      407.9   1  8.8752  0.003327 **
## Residuals      7582.7 165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Fit a model to predict dyadic sexual desire based on relationship duration and partner satisfaction,
# but without accounting for participant-level variance
ctl_APD <- lm(`Dyadic sexual desire (Attractive person)` ~
            `Relationship duration` + `MGSS sexual satisfaction (Partner)`,
            data = dat_ctl)

# Perform an ANOVA test for the CTL APD model
Anova(ctl_APD, type = 3)

## Anova Table (Type III tests)
##
## Response: Dyadic sexual desire (Attractive person)
##
##              Sum Sq Df F value    Pr(>F)
## (Intercept)      737.1   1 11.5389 0.0008541 ***
## 'Relationship duration'      18.2   3  0.0948 0.9628077
## 'MGSS sexual satisfaction (Partner)'      56.5   1  0.8842 0.3484284
## Residuals      10539.5 165
## ---

```

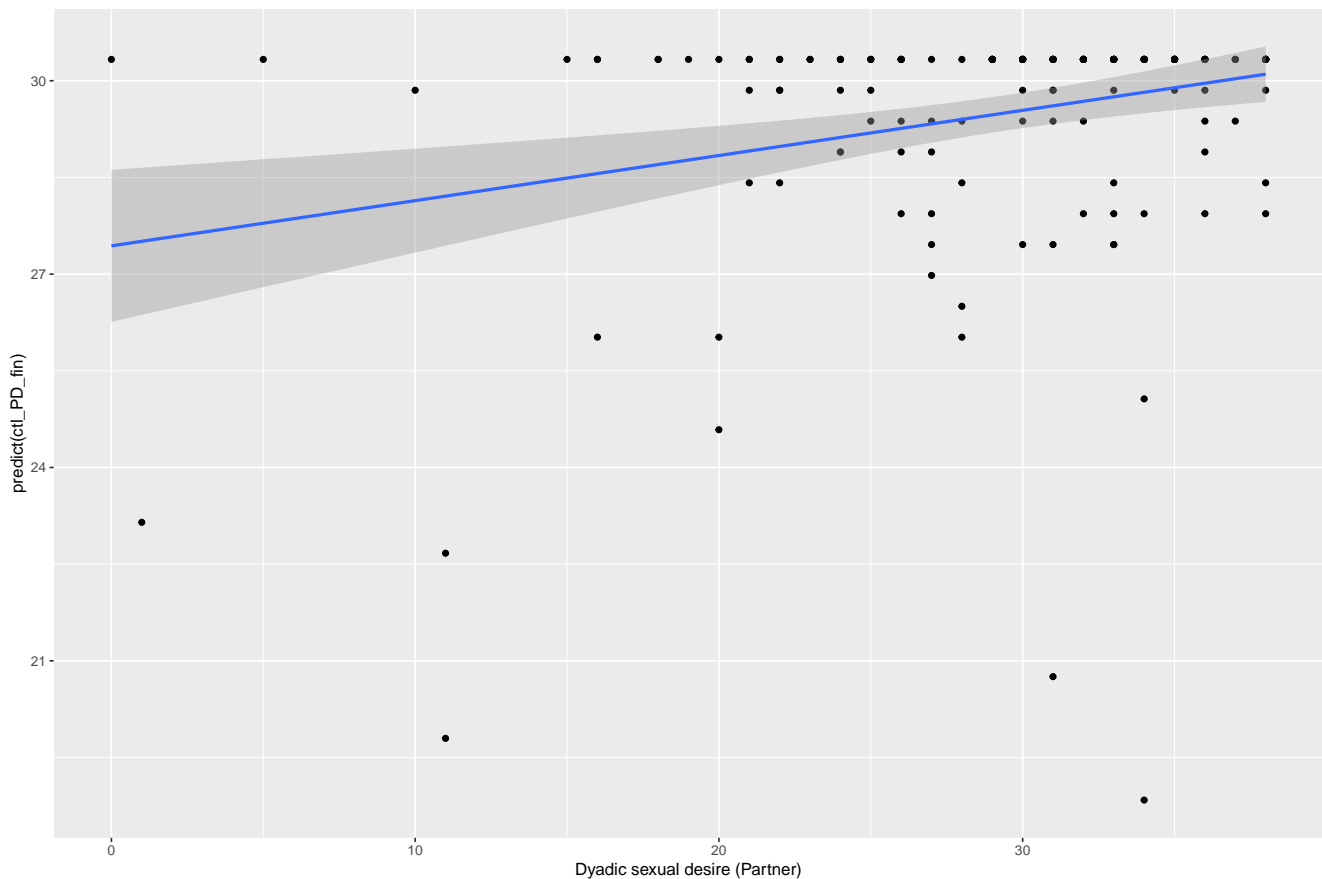
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Create a new dataset with only the desired variables and drop any missing values
dat_tl_PD_fin <- dat_ctl |>
  select(Participant, `Dyadic sexual desire (Partner)`, `MGSS sexual satisfaction (Partner)`) |>
  drop_na()

# Fit a model to predict dyadic sexual desire based on partner satisfaction
ctl_PD_fin <- lm(`Dyadic sexual desire (Partner)` ~ `MGSS sexual satisfaction (Partner)`,
  data = dat_tl_PD_fin)

ggplot(ctl_PD_fin$model, aes(x = `Dyadic sexual desire (Partner)`, y = predict(ctl_PD_fin))) +
  geom_point() +
  geom_smooth(method = "lm")
```

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



```
summary(ctl_PD_fin)
```

```
##
## Call:
## lm(formula = `Dyadic sexual desire (Partner)` ~ `MGSS sexual satisfaction (Partner)`,
##     data = dat_tl_PD_fin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.331  -4.092   1.563   4.669  15.161
##
## Coefficients:
```

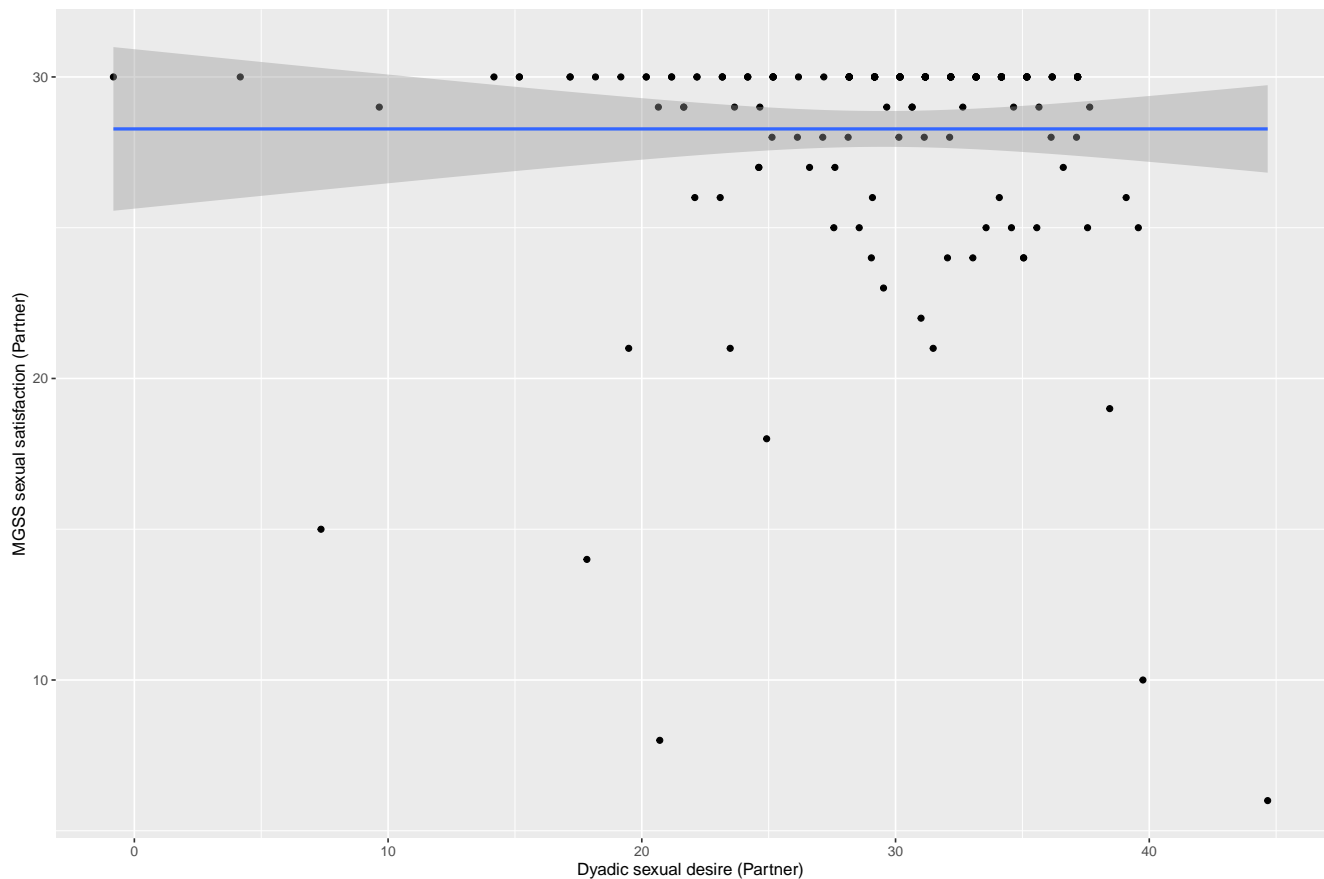


```
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   15.9666      3.8417   4.156 5.15e-05 ***
## 'MGSS sexual satisfaction (Partner)' 0.4788      0.1346   3.558 0.000487 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.844 on 168 degrees of freedom
## Multiple R-squared:  0.07006,    Adjusted R-squared:  0.06452
## F-statistic: 12.66 on 1 and 168 DF,  p-value: 0.0004865
```

```
# Create a new dataset with the predicted values for dyadic sexual desire added back in
dat_ctl <- dat_t1_PD_fin |>
  mutate(`Dyadic sexual desire (Partner)` = mean(`Dyadic sexual desire (Partner)` ) +
    resid(ctl_PD_fin))

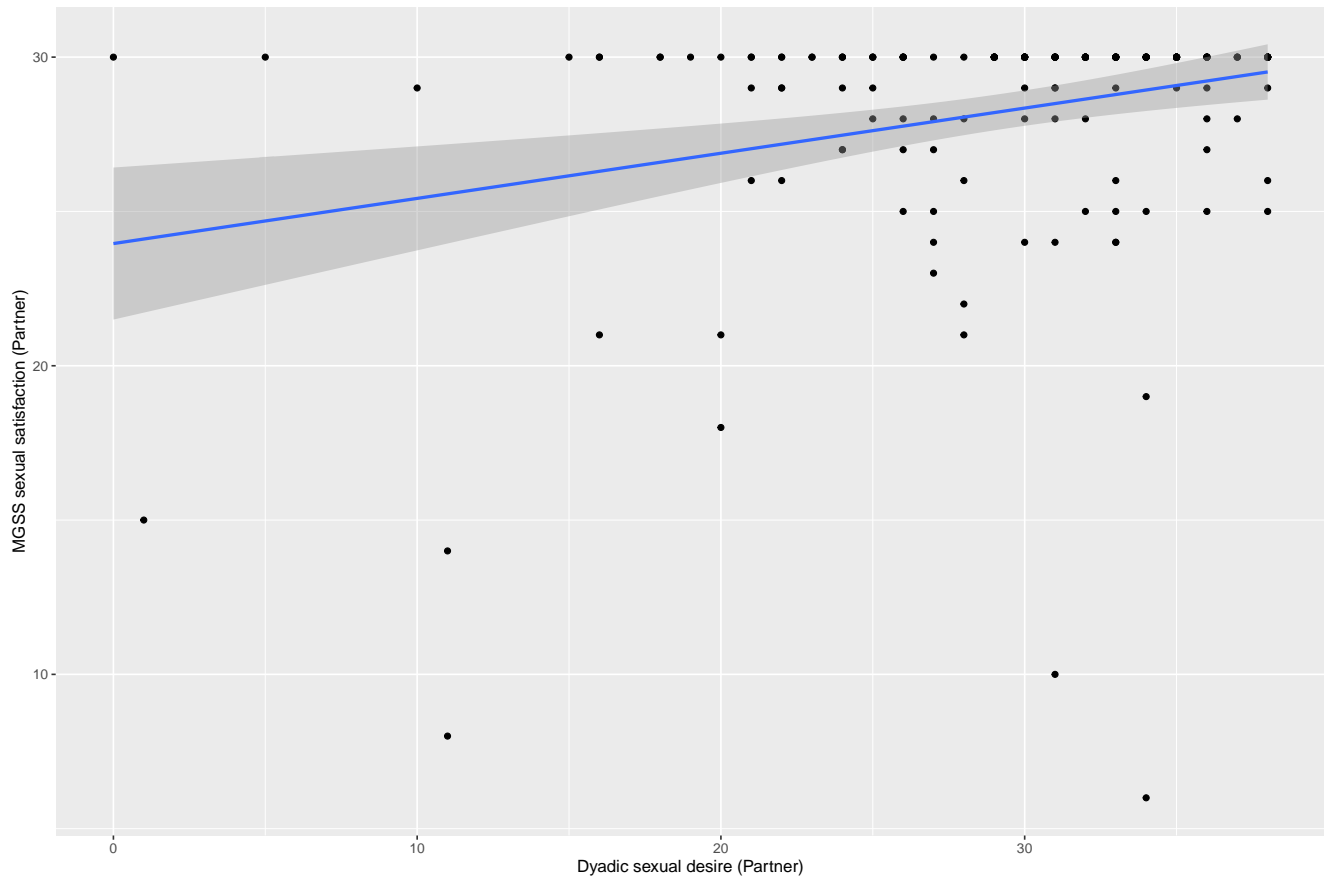
ggplot(dat_ctl, aes(x = `Dyadic sexual desire (Partner)`, y = `MGSS sexual satisfaction (Partner)`)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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



```
ggplot(dat_t1_PD_fin, aes(x = `Dyadic sexual desire (Partner)`, y = `MGSS sexual satisfaction (Partner)`)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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



```
# Convert the dyadic sexual desire variable to a numeric value
dat <- dat |>
  mutate(`Dyadic sexual desire (Partner)` = as.numeric(`Dyadic sexual desire (Partner)`)) |>
  rows_update(dat_ctl |>
    select(~MGSS sexual satisfaction (Partner)`),
    by = "Participant", unmatched = "ignore")
```

4 Hypothesis tests

4.1 Hypothesis 1: All dimensions of trait sexual desire (TSD) will be higher in men than in women, and the differences will be stronger or weaker according to relationship status

We tested whether relationship type and gender interact as predictors of sexual desire (H1a: Solitary TSD; H1b: Dyadic TSD toward an attractive person; H1c: Dyadic TSD toward a partner). To examine this hypothesis, we modeled the effects of relationship type and gender on each of the three TSD scores.

However, models using the original TSD scores did not meet the assumption of normally distributed residuals. To address this, we applied an ordered normalization transformation to each TSD variable. We then fitted and compared models predicting both the original (as a proportion, to make scores comparable) and transformed (normalized) TSD dimensions. In all three cases, models using the normalized variables provided a better fit, so all inferences are based on these models.

4.1.1 Data

A data frame was created with one row per participant, where sexual desire variables were normalized as proportions. An ordered quantile normalization transformation (Peterson & Cavanaugh, 2020) was then applied using the

`orderNorm` function from the `bestNormalize` package (Peterson, 2021), and the transformed values were added as new variables.

```
# Process the dataset and create transformed variables
dat_m1 <- dat |>
  # Group the data by participant
  group_by(Participant) |>
  # Select only the first (top) observation for each participant
  slice_head() |>
  # Remove the grouping structure to avoid unintended behavior in later operations
  ungroup() |>
  # Create new proportion variables by normalizing each sexual desire measure
  mutate("Solitary sexual desire (proportion)" =
    `Solitary sexual desire` / 31,
    "Dyadic sexual desire: Attractive person (proportion)" =
    `Dyadic sexual desire (Attractive person)` / 32,
    "Dyadic sexual desire: Partner (proportion)" =
    `Dyadic sexual desire (Partner)` / 38)

# Apply ordered normalization transformations to the proportion variables
trs_SSD <- orderNorm(dat_m1$`Solitary sexual desire (proportion)`)

## Warning in orderNorm(dat_m1$`Solitary sexual desire (proportion)`): Ties in data, Normal distribution n
trs_DSdat <- orderNorm(dat_m1$`Dyadic sexual desire: Attractive person (proportion)`)

## Warning in orderNorm(dat_m1$`Dyadic sexual desire: Attractive person (proportion)`): Ties in data, Normal
trs_DSdpt <- orderNorm(dat_m1$`Dyadic sexual desire: Partner (proportion)`)

## Warning in orderNorm(dat_m1$`Dyadic sexual desire: Partner (proportion)`): Ties in data, Normal distrib

# Add the transformed variables back into the dataset
dat_m1 <- dat_m1 |>
  mutate("Solitary sexual desire (normalized)" =
    predict(trs_SSD), # Transformed solitary sexual desire
    "Dyadic sexual desire: Attractive person (normalized)" =
    predict(trs_DSdat), # Transformed dyadic sexual desire (attractive person)
    "Dyadic sexual desire: Partner (normalized)" =
    predict(trs_DSdpt)) # Transformed dyadic sexual desire (partner)
```

4.1.2 Hypothesis 1a: Solitary TSD

4.1.2.1 Model the effects of relationship type and gender on Solitary TSD We fitted and checked model fitness of both the original (proportion; `m1a_prop`) and transformed (normalized; `m1a_norm`) TSD scores

```
m1a_prop <- lm(`Solitary sexual desire (proportion)` ~ Gender * Relationship,
  data = dat_m1)

m1a_norm <- lm(`Solitary sexual desire (normalized)` ~ Gender * Relationship,
  data = dat_m1)
```

4.1.2.1.1 Model assumptions Most model assumptions were checked using the `check_model` function from the `performance` package (Lüdtke et al., 2021), and reported in Fig. S3. These assumptions do not include collinearity, as the function plots VIF instead of the recommended Generalized Variance Inflation Factors ($GVIF$) and the most comparable $GVIF^{1/(2 \times df)}$ (Fox & Monette, 1992). Instead, $GVIF$ and $GVIF^{1/(2 \times df)}$ values are reported in Table ??.

Figure S3. Model assumptions

This figure includes most assumptions: linearity, homogeneity of variance, and normality of both residuals and random effects.

```
ggarrange(ggarrange(plotlist = plot(check_model(m1a_prop, panel = FALSE)),
                                ncol = 2, nrow = 3),
          ggarrange(plotlist = plot(check_model(m1a_norm, panel = FALSE)),
                                ncol = 2, nrow = 3),
          labels = "AUTO")

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 0.37096

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.19266

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 5.6133e-15

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 0.036702

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## 0.37096

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.19266

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 5.6133e-15

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.036702

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 0.37096

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.19266

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 5.6133e-15

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 0.036702

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## 0.37096

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.19266

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
```

```

## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 5.6133e-15

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.036702

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at -0.4311

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.67558

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 6.1313e-15

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 0.4513

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## -0.4311

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.67558

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 6.1313e-15

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.4513

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at -0.4311

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.67558

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 6.1313e-15

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 0.4513

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## -0.4311

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.67558

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x

```

```
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 6.1313e-15

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.4513
```

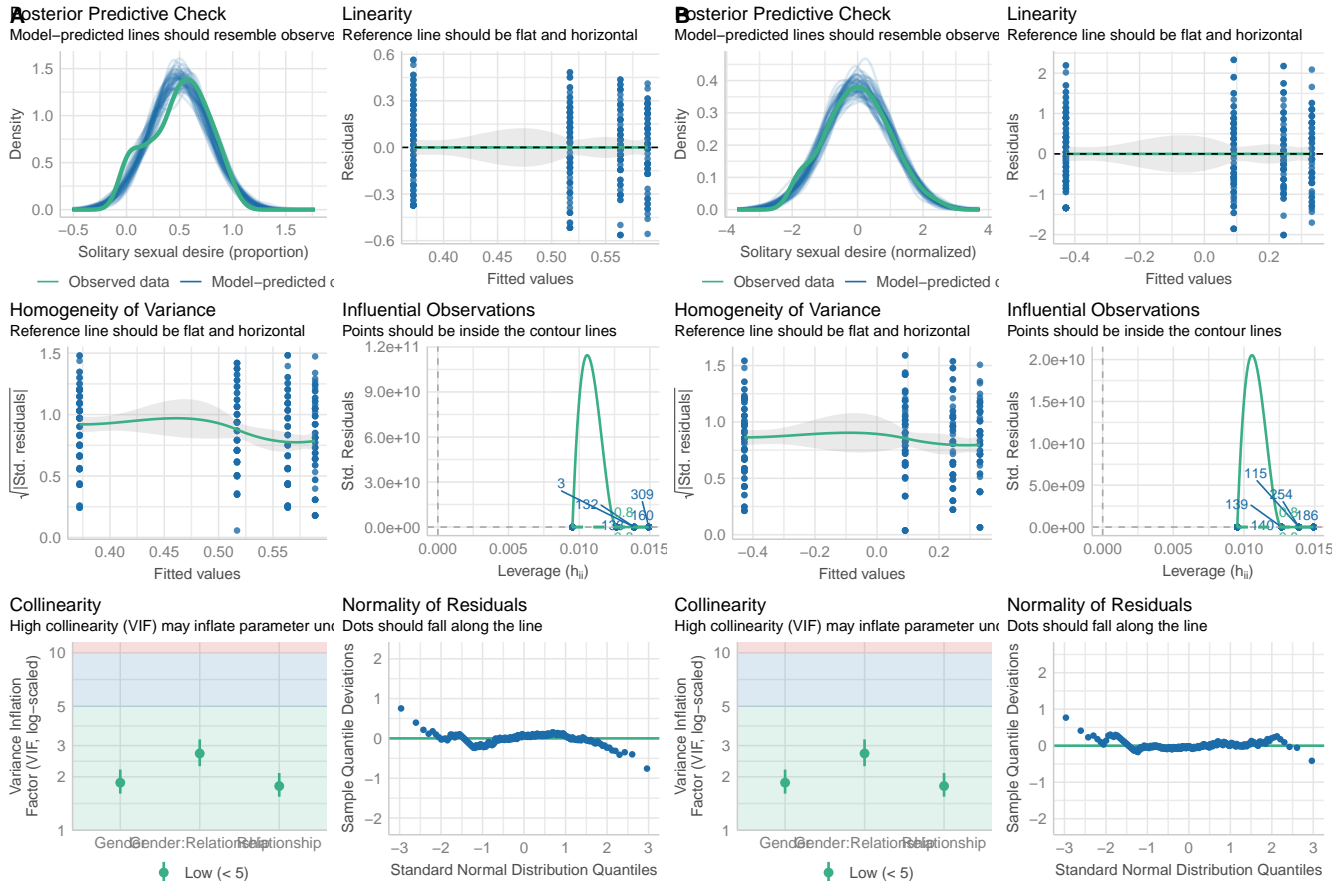


Figure S3. Model assumptions. Plots represent linearity, homogeneity of variance, and normality of both residuals and random effects (as QQ plots), respectively.

5 Session info (for reproducibility)

```
library(pander)
pander(sessionInfo(), locale = FALSE)
```

R version 4.4.2 (2024-10-31)

Platform: x86_64-pc-linux-gnu

attached base packages: stats, graphics, grDevices, utils, datasets, methods and base

other attached packages: pander(v.0.6.5), Hmisc(v.5.2-2), effectsize(v.1.0.0), bestNormalize(v.1.9.1), berry-Functions(v.1.22.5), emmeans(v.1.10.7), lmerTest(v.3.1-3), lme4(v.1.1-36), Matrix(v.1.7-2), scales(v.1.3.0), psych(v.2.4.12), kableExtra(v.1.4.0), performance(v.0.13.0), PerformanceAnalytics(v.2.0.8), quantmod(v.0.4.26), TTR(v.0.24.4), xts(v.0.14.1), zoo(v.1.8-12), tidyquant(v.1.0.10), ggpubr(v.0.6.0), lubridate(v.1.9.4), forcats(v.1.0.0),

stringr(v.1.5.1), *dplyr*(v.1.1.4), *purrr*(v.1.0.4), *readr*(v.2.1.5), *tidyr*(v.1.3.1), *tibble*(v.3.2.1), *ggplot2*(v.3.5.1), *tidyverse*(v.2.0.0), *car*(v.3.1-3), *carData*(v.3.0-5), *ltm*(v.1.2-0), *polycor*(v.0.8-1), *msm*(v.1.8.2), *MASS*(v.7.3-64), *readxl*(v.1.4.3) and *knitr*(v.1.49)

loaded via a namespace (and not attached): *rstudioapi*(v.0.17.1), *datawizard*(v.1.0.0), *magrittr*(v.2.0.3), *TH.data*(v.1.1-3), *estimability*(v.1.5.1), *farver*(v.2.1.2), *nloptr*(v.2.1.1), *rmarkdown*(v.2.29), *vctrs*(v.0.6.5), *minqa*(v.1.2.8), *base64enc*(v.0.1-3), *rstatix*(v.0.7.2), *butcher*(v.0.3.4), *htmltools*(v.0.5.8.1), *curl*(v.6.2.0), *broom*(v.1.0.7), *cellranger*(v.1.1.0), *Formula*(v.1.2-5), *parallelly*(v.1.4.1.0), *htmlwidgets*(v.1.6.4), *sandwich*(v.3.1-1), *admisc*(v.0.37), *lifecycle*(v.1.0.4), *iterators*(v.1.0.14), *pkgconfig*(v.2.0.3), *R6*(v.2.5.1), *fastmap*(v.1.2.0), *rbibutils*(v.2.3), *future*(v.1.34.0), *digest*(v.0.6.37), *numDeriv*(v.2016.8-1.1), *colorspace*(v.2.1-1), *labeling*(v.0.4.3), *timechange*(v.0.3.0), *mgcv*(v.1.9-1), *abind*(v.1.4-8), *compiler*(v.4.4.2), *rngtools*(v.1.5.2), *withr*(v.3.0.2), *doParallel*(v.1.0.17), *htmlTable*(v.2.4.3), *backports*(v.1.5.0), *ggsignif*(v.0.6.4), *lava*(v.1.8.1), *tools*(v.4.4.2), *foreign*(v.0.8-88), *RobStatTM*(v.1.0.11), *future.apply*(v.1.11.3), *nnet*(v.7.3-20), *glue*(v.1.8.0), *quadprog*(v.1.5-8), *nlme*(v.3.1-167), *grid*(v.4.4.2), *checkmate*(v.2.3.2), *cluster*(v.2.1.8), *see*(v.0.10.0), *generics*(v.0.1.3), *recipes*(v.1.1.0), *gtable*(v.0.3.6), *nortest*(v.1.0-4), *tzdb*(v.0.4.0), *class*(v.7.3-23), *data.table*(v.1.16.4), *hms*(v.1.1.3), *xmll2*(v.1.3.6), *ggrepel*(v.0.9.6), *foreach*(v.1.5.2), *pillar*(v.1.10.1), *splines*(v.4.4.2), *lattice*(v.0.22-6), *survival*(v.3.8-3), *tidyselect*(v.1.2.1), *gridExtra*(v.2.3), *reformulas*(v.0.4.0), *bookdown*(v.0.42), *svglite*(v.2.1.3), *xfun*(v.0.50), *expm*(v.1.0-0), *hardhat*(v.1.4.0), *timeDate*(v.4041.110), *stringi*(v.1.8.4), *yaml*(v.2.3.10), *boot*(v.1.3-31), *evaluate*(v.1.0.3), *codetools*(v.0.2-20), *cli*(v.3.6.3), *rpart*(v.4.1.24), *parameters*(v.0.24.1), *xtable*(v.1.8-4), *systemfonts*(v.1.2.1), *Rdpack*(v.2.6.2), *munsell*(v.0.5.1), *Rcpp*(v.1.0.14), *globals*(v.0.16.3), *coda*(v.0.19-4.1), *parallel*(v.4.4.2), *gower*(v.1.0.2), *bayes-testR*(v.0.15.1), *doRNG*(v.1.8.6.1), *listenv*(v.0.9.1), *viridisLite*(v.0.4.2), *mvtnorm*(v.1.3-3), *ipred*(v.0.9-15), *prodim*(v.2024.06.25), *insight*(v.1.0.1), *rlang*(v.1.1.5), *cowplot*(v.1.1.3), *multcomp*(v.1.4-28) and *mnormt*(v.2.1.1)

6 Supplementary references

- Fox, J., & Monette, G. (1992). Generalized Collinearity Diagnostics. *Journal of the American Statistical Association*, 87(417), 178–183. <https://doi.org/10.1080/01621459.1992.10475190>
- Kaufman, D., & Sweet, R. (1974). Contrast Coding in Least Squares Regression Analysis. *American Educational Research Journal*, 11(4), 359–377. <https://doi.org/10.3102/00028312011004359>
- Keppel, G., & Zedeck, S. (1989). *Data analysis for research designs: Analysis of variance and multiple regression/correlation approaches*. W.H. Freeman.
- Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13), 1–26. <https://doi.org/10.18637/jss.v082.i13>
- Lenth, R. V. (2023). *Emmeans: Estimated marginal means, aka least-squares means* [R package version 1.8.9]. <https://CRAN.R-project.org/package=emmeans>
- Long, J. A. (2019). *Interactions: Comprehensive, user-friendly toolkit for probing interactions* [R package version 1.1.0]. <https://cran.r-project.org/package=interactions>
- Lüdtke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). performance: An R package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, 6(60), 3139. <https://doi.org/10.21105/joss.03139>
- Peterson, R. A. (2021). Finding Optimal Normalizing Transformations via bestNormalize. *The R Journal*, 13(1), 310. <https://doi.org/10.32614/RJ-2021-041>
- Peterson, R. A., & Cavanaugh, J. E. (2020). Ordered quantile normalization: A semiparametric transformation built for the cross-validation era. *Journal of Applied Statistics*, 47(13–15), 2312–2327. <https://doi.org/10.1080/02664763.2019.1630372>
- Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, 17(5), 1–25. <https://doi.org/10.18637/jss.v017.i05>
- Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>
- Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation* [R package version 1.1.3]. <https://CRAN.R-project.org/package=dplyr>
- Xie, Y. (2014). Knitr: A comprehensive tool for reproducible research in R [ISBN 978-1466561595]. In V. Stodden, F. Leisch & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781315373461-1>
- Zhu, H. (2021). *Kableextra: Construct complex table with 'kable' and pipe syntax* [R package version 1.3.4]. <https://CRAN.R-project.org/package=kableExtra>