

# How do experiences of violence affect women's preferences for facial masculinity according to resource availability? An exploratory study using eye-tracking

## Code and analyses

Milena Vásquez-Amézquita <sup>3,1,\*</sup> Wendy Medina-Sarmiento<sup>1</sup> Valentina Cepeda<sup>1</sup>  
Andrés Castellanos-Chacón  Marina Begoña Martínez-González   
Juan David Leongómez 

25 September, 2024

<sup>1</sup> Faculty of Psychology, Universidad El Bosque, Bogota, Colombia.

<sup>2</sup> EvoCo: Human Behaviour and Evolution Lab, Faculty of Psychology, Universidad El Bosque, Bogota, Colombia

<sup>3</sup> Faculty of Psychology, Universidad de la Costa, Barranquilla, Colombia.

\* Correspondence: [mvasquezam@unbosque.edu.co](mailto: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., Castellanos-Chacón, A., Medina-Sarmiento, W., Cepeda, V., Martínez-González, M. B., & Leongómez, J. D., (in prep). *How do experiences of violence affect women's preferences for facial masculinity according to resource availability? An exploratory study using eye-tracking*.

Data available from the Open Science Framework (OSF): <https://doi.org/10.XXXXXX/OSF.IO/XX XXX>. 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 L<sup>A</sup>T<sub>E</sub>X.

---

## Contents

<b>1 Preliminaries</b>	<b>4</b>
1.1 Load packages . . . . .	4
1.2 Custom functions . . . . .	4
1.2.1 pval.levand small.lev . . . . .	4
1.2.2 corr.stars . . . . .	5
1.2.3 lmer.anova.tab, glmer.anova.tab and lm.anova.tab . . . . .	6
1.2.4 main.eff.contr . . . . .	8
1.2.5 inter.contr . . . . .	9
1.2.6 full.contr . . . . .	10
1.2.7 mods.comp . . . . .	11
1.2.8 plot.exp . . . . .	12
1.2.9 plot.best.mod . . . . .	13
1.3 Independent stimuli evaluation . . . . .	14
1.3.1 Masculinity ratings . . . . .	14

1.3.2	Age ratings . . . . .	16
1.3.2.1	Histogram of perceived age . . . . .	16
1.4	Load and wrangle main experiment data . . . . .	17
1.4.1	Individual databases (by data type/source) . . . . .	17
1.4.1.1	Eye-tracking data . . . . .	18
1.4.1.2	Questionnaires . . . . .	18
1.4.1.2.1	Principal component analysis (PCA) . . . . .	23
1.4.1.2.2	Clean questionnaire data . . . . .	27
1.4.1.3	Subjective evaluation of stimuli . . . . .	28
1.4.1.3.1	Wide format . . . . .	28
1.4.1.3.2	Long format . . . . .	29
1.4.1.4	Resource availability . . . . .	30
1.4.2	Full, final database . . . . .	30
1.4.2.1	Join data files . . . . .	30
1.4.2.2	Filtered database . . . . .	30
1.4.3	Final individual databases filtered to the final sample . . . . .	30
1.4.3.1	Resource availability (filtered) . . . . .	30
1.4.3.2	Questionnaires (filtered) . . . . .	31
<b>2</b>	<b>Descriptives</b>	<b>31</b>
2.1	Number and age of participants in each condition . . . . .	31
2.2	Select and wrangle data for descriptive plots . . . . .	32
2.3	Distribution of values across variables . . . . .	33
2.3.1	Sociodemographic variables . . . . .	33
2.3.2	Access to resources . . . . .	34
2.3.3	Health-related variables . . . . .	35
2.3.4	Food security . . . . .	36
2.3.5	Hormonal variables . . . . .	38
2.3.6	Self-perceived conditions . . . . .	39
2.3.7	Current/last partner perception . . . . .	40
2.3.8	Context violence . . . . .	41
2.3.9	Gender and partner violence . . . . .	43
2.3.10	Subjective evaluation of stimuli . . . . .	44
2.4	Correlations . . . . .	45
2.4.1	Correlations between partner violence and responses to masculinized and feminized stimuli . . . . .	45
<b>3</b>	<b>Manipulation check</b>	<b>48</b>
3.1	Resource availability dimensions by condition . . . . .	48
3.2	Effect of sexual dimorphism manipulation on masculinity and attractiveness ratings, by condition . . . . .	49
<b>4</b>	<b>Models of the experimental design</b>	<b>51</b>
4.1	Duration of First Fixations (DFF) . . . . .	51
4.1.1	Data . . . . .	51
4.1.2	Fit linear mixed model . . . . .	52
4.1.2.1	Model assumptions . . . . .	52
4.1.3	Table of fixed effects . . . . .	52
4.1.4	Estimated marginal means and <i>post-hoc</i> contrasts of significant effects . . . . .	53
4.1.4.1	Main effect: Sexual dimorphism . . . . .	53
4.1.5	Full design contrasts . . . . .	53
4.1.6	Figure for the DFF model . . . . .	54
4.2	Total Fixation Duration (TFD) . . . . .	54
4.2.1	Data . . . . .	54
4.2.2	Fit linear mixed model . . . . .	55
4.2.2.1	Model assumptions . . . . .	55
4.2.3	Table of fixed effects . . . . .	56
4.2.4	Estimated marginal means and <i>post-hoc</i> contrasts of significant effects . . . . .	57
4.2.4.1	Main effect: Sexual dimorphism . . . . .	57

4.2.4.2	Interaction: Relationship × Sexual dimorphism . . . . .	57
4.2.5	Full design contrasts . . . . .	57
4.2.6	Figure for the TFD model . . . . .	58
4.3	Number of Fixations (NF) . . . . .	58
4.3.1	Data . . . . .	58
4.3.2	Fit linear mixed model . . . . .	59
4.3.2.1	Model assumptions . . . . .	59
4.3.3	Table of fixed effects . . . . .	60
4.3.4	Estimated marginal means and <i>post-hoc</i> contrasts of significant effects . . . . .	60
4.3.4.1	Main effect: Sexual dimorphism . . . . .	60
4.3.4.2	Interaction: Relationship × Sexual dimorphism . . . . .	60
4.3.5	Full design contrasts . . . . .	61
4.3.6	Figure for the NF model . . . . .	61
4.4	Proportion of Chosen Faces (PCF) . . . . .	62
4.4.1	Data . . . . .	62
4.4.2	Fit linear mixed model . . . . .	62
4.4.2.1	Model assumptions . . . . .	62
4.4.3	Table of fixed effects . . . . .	63
4.4.4	Estimated marginal means and <i>post-hoc</i> contrasts of significant effects . . . . .	64
4.4.4.1	Main effect: Sexual dimorphism . . . . .	64
4.4.4.2	Interaction: Relationship × Sexual dimorphism . . . . .	64
4.4.5	Full design contrasts . . . . .	65
4.4.6	Figure for the NF model . . . . .	65
<b>5</b>	<b>Exploring interactions with covariates</b>	<b>66</b>
5.1	Duration of First Fixations (DFF) . . . . .	66
5.1.1	Fit linear models with interactions partner violence variables . . . . .	66
5.1.2	Model comparison . . . . .	66
5.1.3	Best model . . . . .	67
5.1.3.1	Table of fixed effects . . . . .	67
5.1.3.2	Figure for the best model predicting DFF . . . . .	68
5.2	Total Fixation Duration (TFD) . . . . .	68
5.2.1	Fit linear models with interactions partner violence variables . . . . .	68
5.2.2	Model comparison . . . . .	69
5.2.3	Best model . . . . .	69
5.2.3.1	Table of fixed effects . . . . .	70
5.2.3.2	Figure for the best model predicting TFD . . . . .	71
5.3	Number of Fixations (NF) . . . . .	71
5.3.1	Fit linear models with interactions partner violence variables . . . . .	71
5.3.2	Model comparison . . . . .	72
5.3.3	Best model . . . . .	72
5.3.3.1	Table of fixed effects . . . . .	73
5.3.3.2	Figure for the best model predicting NF . . . . .	74
5.4	Proportion of Chosen Faces (PCF) . . . . .	74
5.4.1	Fit linear models with interactions partner violence variables . . . . .	74
5.4.2	Model comparison . . . . .	75
5.4.3	Best model . . . . .	75
5.4.3.1	Table of fixed effects . . . . .	76
5.4.3.2	Figure for the best model predicting PCF . . . . .	77
<b>6</b>	<b>Final figures and tables (included in the main paper)</b>	<b>77</b>
6.1	Table of fixed effects for experimental design models . . . . .	77
6.2	Figure of experimental design models . . . . .	78
6.3	Table of fixed effects for best-supported models . . . . .	79
6.4	Figure of experimental design models . . . . .	80
<b>7</b>	<b>Session info (for reproducibility)</b>	<b>81</b>

# 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, 2020).

Linear mixed models were fitted using `lmerTest` (Kuznetsova et al., 2017), assumptions were performed using `performance` (Lüdecke et al., 2021), contrasts and interactions were explored using `emmeans` (Lenth, 2024).

Used packages also include `osfr` (Wolen et al., 2020) to download and open data files directly from the Open Science Framework ([OSF](#)), using the `osf_retrieve_file` and `osf_download` functions.

All packages used in this file can be directly installed from the Comprehensive R Archive Network ([CRAN](#)). For a complete list of packages used to create this file, and their versions, see section 7, at the end of the document.

```
library(car)
library(MASS)
library(ggstats)
library(tidyverse)
library(ggpubr)
library(readxl)
library(lmerTest)
library(emmeans)
library(knitr)
library(kableExtra)
library(performance)
library(GGally)
library(scales)
library(factoextra)
library(FactoMineR)
library(gtools)
library(bbmle)
library(effectsize)
library(insight)
```

## 1.2 Custom functions

### 1.2.1 `pval.lev` and `small.lev`

This functions take small numbers and format them in L<sup>A</sup>T<sub>E</sub>X. For  $p$  values, significant results are highlighted in bold.

```
# Define a function 'pval.lev' to format p-values based on specific thresholds.
pval.lev <- function(pvals) {
  # If the p-value is less than 0.0001, return the string '\textbf{< 0.0001}'.
  ifelse(pvals < 0.0001,
    "\\textbf{< 0.0001}",
    # If the p-value is less than 0.001, return the string '\textbf{< 0.001}'.
    ifelse(pvals < 0.001,
      "\\textbf{< 0.001}",
      # If the p-value is less than 0.05, format it with bold text and round to 4
      # decimal places.
      ifelse(pvals < 0.05,
        paste0("\\textbf{", round(pvals, 4), "}"),
        # If the p-value is greater than or equal to 0.05, return the string 'p < 0.05'.
        ifelse(pvals >= 0.05,
          "p < 0.05",
          # If the p-value is greater than or equal to 0.1, return the string 'p < 0.1'.
          ifelse(pvals >= 0.1,
            "p < 0.1",
            # If the p-value is greater than or equal to 0.5, return the string 'p < 0.5'.
            ifelse(pvals >= 0.5,
              "p < 0.5",
              # If the p-value is greater than or equal to 1, return the string 'p < 1'.
              ifelse(pvals >= 1,
                "p < 1",
                # If the p-value is greater than or equal to 10, return the string 'p < 10'.
                ifelse(pvals >= 10,
                  "p < 10",
                  # If the p-value is greater than or equal to 100, return the string 'p < 100'.
                  ifelse(pvals >= 100,
                    "p < 100",
                    # If the p-value is greater than or equal to 1000, return the string 'p < 1000'.
                    ifelse(pvals >= 1000,
                      "p < 1000",
                      # If the p-value is greater than or equal to 10000, return the string 'p < 10000'.
                      ifelse(pvals >= 10000,
                        "p < 10000",
                        # If the p-value is greater than or equal to 100000, return the string 'p < 100000'.
                        ifelse(pvals >= 100000,
                          "p < 100000",
                          # If the p-value is greater than or equal to 1000000, return the string 'p < 1000000'.
                          ifelse(pvals >= 1000000,
                            "p < 1000000",
                            # If the p-value is greater than or equal to 10000000, return the string 'p < 10000000'.
                            ifelse(pvals >= 10000000,
                              "p < 10000000",
                              # If the p-value is greater than or equal to 100000000, return the string 'p < 100000000'.
                              ifelse(pvals >= 100000000,
                                "p < 100000000",
                                # If the p-value is greater than or equal to 1000000000, return the string 'p < 1000000000'.
                                ifelse(pvals >= 1000000000,
                                  "p < 1000000000",
                                  # If the p-value is greater than or equal to 10000000000, return the string 'p < 10000000000'.
                                  ifelse(pvals >= 10000000000,
                                    "p < 10000000000",
                                    # If the p-value is greater than or equal to 100000000000, return the string 'p < 100000000000'.
                                    ifelse(pvals >= 100000000000,
                                      "p < 100000000000",
                                      # If the p-value is greater than or equal to 1000000000000, return the string 'p < 1000000000000'.
                                      ifelse(pvals >= 1000000000000,
                                        "p < 1000000000000",
                                        # If the p-value is greater than or equal to 10000000000000, return the string 'p < 10000000000000'.
                                        ifelse(pvals >= 10000000000000,
                                          "p < 10000000000000",
                                          # If the p-value is greater than or equal to 100000000000000, return the string 'p < 100000000000000'.
                                          ifelse(pvals >= 100000000000000,
                                            "p < 100000000000000",
                                            # If the p-value is greater than or equal to 1000000000000000, return the string 'p < 1000000000000000'.
                                            ifelse(pvals >= 1000000000000000,
                                              "p < 1000000000000000",
                                              # If the p-value is greater than or equal to 10000000000000000, return the string 'p < 10000000000000000'.
                                              ifelse(pvals >= 10000000000000000,
                                                "p < 10000000000000000",
                                                # If the p-value is greater than or equal to 100000000000000000, return the string 'p < 100000000000000000'.
                                                ifelse(pvals >= 100000000000000000,
                                                  "p < 100000000000000000",
                                                  # If the p-value is greater than or equal to 1000000000000000000, return the string 'p < 1000000000000000000'.
                                                  ifelse(pvals >= 1000000000000000000,
                                                    "p < 1000000000000000000",
                                                    # If the p-value is greater than or equal to 10000000000000000000, return the string 'p < 10000000000000000000'.
                                                    ifelse(pvals >= 10000000000000000000,
                                                      "p < 10000000000000000000",
                                                      # If the p-value is greater than or equal to 100000000000000000000, return the string 'p < 100000000000000000000'.
                                                      ifelse(pvals >= 100000000000000000000,
                                                        "p < 100000000000000000000",
                                                        # If the p-value is greater than or equal to 1000000000000000000000, return the string 'p < 1000000000000000000000'.
                                                        ifelse(pvals >= 1000000000000000000000,
                                                          "p < 1000000000000000000000",
                                                          # If the p-value is greater than or equal to 10000000000000000000000, return the string 'p < 10000000000000000000000'.
                                                          ifelse(pvals >= 10000000000000000000000,
                                                            "p < 10000000000000000000000",
                                                            # If the p-value is greater than or equal to 100000000000000000000000, return the string 'p < 100000000000000000000000'.
                                                            ifelse(pvals >= 100000000000000000000000,
                                                              "p < 100000000000000000000000",
                                                              # If the p-value is greater than or equal to 1000000000000000000000000, return the string 'p < 1000000000000000000000000'.
                                                              ifelse(pvals >= 1000000000000000000000000,
                                                                "p < 1000000000000000000000000",
                                                                # If the p-value is greater than or equal to 10000000000000000000000000, return the string 'p < 10000000000000000000000000'.
                                                                ifelse(pvals >= 10000000000000000000000000,
                                                                  "p < 10000000000000000000000000",
                                                                  # If the p-value is greater than or equal to 100000000000000000000000000, return the string 'p < 100000000000000000000000000'.
                                                                  ifelse(pvals >= 100000000000000000000000000,
                                                                    "p < 100000000000000000000000000",
                                                                    # If the p-value is greater than or equal to 1000000000000000000000000000, return the string 'p < 1000000000000000000000000000'.
                                                                    ifelse(pvals >= 1000000000000000000000000000,
                                                                      "p < 1000000000000000000000000000",
                                                                      # If the p-value is greater than or equal to 10000000000000000000000000000, return the string 'p < 10000000000000000000000000000'.
                                                                      ifelse(pvals >= 10000000000000000000000000000,
                                                                        "p < 10000000000000000000000000000",
                                                                        # If the p-value is greater than or equal to 100000000000000000000000000000, return the string 'p < 100000000000000000000000000000'.
                                                                        ifelse(pvals >= 100000000000000000000000000000,
                                                                          "p < 100000000000000000000000000000",
                                                                          # If the p-value is greater than or equal to 1000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000'.
                                                                          ifelse(pvals >= 1000000000000000000000000000000,
                                                                            "p < 1000000000000000000000000000000",
                                                                            # If the p-value is greater than or equal to 10000000000000000000000000000000, return the string 'p < 10000000000000000000000000000000'.
                                                                            ifelse(pvals >= 10000000000000000000000000000000,
                                                                              "p < 10000000000000000000000000000000",
                                                                              # If the p-value is greater than or equal to 100000000000000000000000000000000, return the string 'p < 100000000000000000000000000000000'.
                                                                              ifelse(pvals >= 100000000000000000000000000000000,
                                                                                "p < 100000000000000000000000000000000",
                                                                                # If the p-value is greater than or equal to 1000000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000000'.
                                                                                ifelse(pvals >= 1000000000000000000000000000000000,
                                                                                  "p < 1000000000000000000000000000000000",
                                                                                  # If the p-value is greater than or equal to 10000000000000000000000000000000000, return the string 'p < 10000000000000000000000000000000000'.
                                                                                  ifelse(pvals >= 10000000000000000000000000000000000,
                                                                                    "p < 10000000000000000000000000000000000",
                                                                                    # If the p-value is greater than or equal to 100000000000000000000000000000000000, return the string 'p < 100000000000000000000000000000000000'.
                                                                                    ifelse(pvals >= 100000000000000000000000000000000000,
                                                                                      "p < 100000000000000000000000000000000000",
                                                                                      # If the p-value is greater than or equal to 1000000000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000000000'.
                                                                                      ifelse(pvals >= 1000000000000000000000000000000000000,
                                                                                        "p < 1000000000000000000000000000000000000",
                                                                                        # If the p-value is greater than or equal to 10000000000000000000000000000000000000, return the string 'p < 10000000000000000000000000000000000000'.
                                                                                        ifelse(pvals >= 10000000000000000000000000000000000000,
                                                                                          "p < 10000000000000000000000000000000000000",
                                                                                          # If the p-value is greater than or equal to 100000000000000000000000000000000000000, return the string 'p < 100000000000000000000000000000000000000'.
                                                                                          ifelse(pvals >= 100000000000000000000000000000000000000,
                                                                                            "p < 100000000000000000000000000000000000000",
                                                                                            # If the p-value is greater than or equal to 1000000000000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000000000000'.
                                                                                            ifelse(pvals >= 1000000000000000000000000000000000000000,
                                                                                              "p < 1000000000000000000000000000000000000000",
                                                                                              # If the p-value is greater than or equal to 10000000000000000000000000000000000000000, return the string 'p < 10000000000000000000000000000000000000000'.
                                                                                              ifelse(pvals >= 10000000000000000000000000000000000000000,
                                                                                                "p < 10000000000000000000000000000000000000000",
                                                                                                # If the p-value is greater than or equal to 100000000000000000000000000000000000000000, return the string 'p < 100000000000000000000000000000000000000000'.
                                                                                                ifelse(pvals >= 100000000000000000000000000000000000000000,
                                                                                                  "p < 100000000000000000000000000000000000000000",
                                                                                                  # If the p-value is greater than or equal to 1000000000000000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000000000000000'.
                                                                                                  ifelse(pvals >= 1000000000000000000000000000000000000000000,
                                                                                                    "p < 1000000000000000000000000000000000000000000",
                                                                                                    # If the p-value is greater than or equal to 10000000000000000000000000000000000000000000, return the string 'p < 10000000000000000000000000000000000000000000'.
                                                                                                    ifelse(pvals >= 10000000000000000000000000000000000000000000,
                                                                                                      "p < 10000000000000000000000000000000000000000000",
                                                                                                      # If the p-value is greater than or equal to 100000000000000000000000000000000000000000000, return the string 'p < 100000000000000000000000000000000000000000000'.
                                                                                                      ifelse(pvals >= 100000000000000000000000000000000000000000000,
                                                                                                        "p < 100000000000000000000000000000000000000000000",
                                                                                                        # If the p-value is greater than or equal to 1000000000000000000000000000000000000000000000, return the string 'p < 1000000000000000000000000000000000000000000000'.
                                                                                                        ifelse(pvals >= 1000000000000000000000000000000000000000000000,
                                                                                                          "p < 1000000000000000000000000000000000000000000000",
                                          
```

```

        # Otherwise, round the p-value to 2 decimal places.
        round(pvals, 2)
    )
}
}

small.lev <- function(vals) {
  # If the p-value is less than 0.00001, return the string '\textbf{< 0.0001}'.
  ifelse(vals < 0.00001,
    "< 0.00001",
    # If the p-value is less than 0.0001, return the string '\textbf{< 0.0001}'.
    ifelse(vals < 0.0001,
      "< 0.0001",
      # If the p-value is less than 0.001, return the string '\textbf{< 0.001}'.
      ifelse(vals < 0.001,
        "< 0.001",
        # If the p-value is less than 0.05, format it with bold text and round to 4
        # decimal places.
        ifelse(vals < 0.05,
          paste0("", round(vals, 4), ""),
          # Otherwise, round the p-value to 2 decimal places.
          round(vals, 2)
        )
      )
    )
}
}

```

### 1.2.2 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) {
  # Load the 'Hmisc' package, which is required for the 'rcorr' function.
  require(Hmisc)
  # Convert the input 'x' to a matrix in case it is not already.
  x <- as.matrix(x)
  # Compute the correlation matrix (R) and p-values (p) using the 'rcorr' function.
  R <- rcorr(x)$r # Correlation matrix
  p <- rcorr(x)$P # p-value matrix
  # Define significance levels for the stars notation.
  # *** for p < 0.001, ** for p < 0.01, * for p < 0.05, and † for p < 0.10.
  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 'Rnew' that contains the correlations and their significance stars.
Rnew <- matrix(mystars,
                 ncol = ncol(x))
# Ensure the new matrix has the same number of columns as 'x'
# Add the correlation values without stars to the diagonal (self-correlations).
diag(Rnew) <- paste(diag(R), " ",
                     sep = "")
# Set row names and column names of the matrix 'Rnew' to match those of the original matrix.
rownames(Rnew) <- colnames(x)
colnames(Rnew) <- paste(colnames(x), "", sep = "")
# Remove the upper triangle and the diagonal of the matrix to avoid duplication.
Rnew <- as.matrix(Rnew)
Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
# Convert the matrix to a data frame for easier handling.
Rnew <- as.data.frame(Rnew)
# Remove the last column (empty column from the upper triangle) and return the result.
Rnew <- cbind(Rnew[1:length(Rnew) - 1])
# Return the final correlation matrix with significance stars.
return(Rnew)
}

```

### 1.2.3 lmer.anova.tab, glmer.anova.tab and lm.anova.tab

These functions take a model, and creates an ANOVA-type table of fixed effects, formatted in L<sup>A</sup>T<sub>E</sub>X.

```

# lmer.anova.tab for linear mixed-effects models
lmer.anova.tab <- function(model) {

  r2 <- r2_nakagawa(model)

  tab <- anova(model) |>
    rownames_to_column(var = "Effect") |>
    mutate(Effect = str_replace_all(Effect, "_", " "),
           Effect = str_replace_all(Effect, ":", " × "),
           Effect = str_replace_all(Effect, "Freq partner physical", "Physical"),
           df = paste0(NumDF, ", ", round(DenDF, 2))) |>
    rename("F" = "F value",
           "p" = "Pr(>F)") |>
    mutate(F = round(F, 2),
           p = pval.lev(p)) |>
    select(Effect, F, df, p) |>
    add_row(Effect = paste0("Conditional = ", round(r2$R2_conditional, 2))) |>
    add_row(Effect = paste0("Marginal = ", signif(r2$R2_marginal, 2))) |>
    rename("Fixed effect" = "Effect",
           "$F$" = "F",
           "$df$" = "df",
           "$p$" = "p")

  caption <- paste0("ANOVA-type table of fixed effects for the ", find_response(model), " model")

  n_rows <- dim(tab)[1]

  taa <- kable(tab,
                digits = 2,
                booktabs = TRUE,

```

```

    align = c("l", rep("c", 3)),
    linesep = "",
    caption = caption,
    escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
pack_rows("Nakagawa's $R^2$",
          start_row = n_rows-1, end_row = n_rows,
          hline_after = TRUE, hline_before = TRUE,
          escape = FALSE)
return(list(tab = tab, kab = taa))
}

# glmer.anova.tab for generalized linear mixed-effects models
glmer.anova.tab <- function(model) {

r2 <- r2_nakagawa(model)

tab <- Anova(model, type = "II") |>
rownames_to_column(var = "Effect") |>
mutate(Effect = str_replace_all(Effect, "_", " "),
       Effect = str_replace_all(Effect, ":", " × "),
       Effect = str_replace_all(Effect, "Freq partner physical", "Physical")) |>
filter(Effect != "(Intercept)") |>
rename("p" = "Pr(>Chisq)") |>
mutate(Chisq = round(Chisq, 2),
       p = pval.lev(p)) |>
add_row(Effect = paste0("Conditional = ", round(r2$R2_conditional, 2))) |>
add_row(Effect = paste0("Marginal = ", signif(r2$R2_marginal, 2))) |>
rename("Fixed effect" = "Effect",
      "$\\chi^2$" = "Chisq",
      "$df$" = "Df",
      "$p$" = "p")

caption <- paste0("ANOVA-type table of fixed effects for the ", find_response(model), " model")

n_rows <- dim(tab)[1]

taa <- kable(tab,
             digits = 2,
             booktabs = TRUE,
             align = c("l", rep("c", 3)),
             linesep = "",
             caption = caption,
             escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
pack_rows("Nakagawa's $R^2$",
          start_row = n_rows-1, end_row = n_rows,
          hline_after = TRUE, hline_before = TRUE,
          escape = FALSE)
return(list(tab = tab, kab = taa))
}

# lm.anova.tab for linear models
lm.anova.tab <- function(model) {

r2 <- r2(model)

```

```

DenDF <- last(Anova(model, type = 3)$Df)

omega <- data.frame(omega_squared(model)) |>
  rename(Effect = Parameter)

tab <- anova(model) |>
  rownames_to_column(var = "Effect") |>
  filter(Effect != "(Intercept)") |>
  mutate(Effect = str_replace_all(Effect, "_", " "),
         Effect = str_replace_all(Effect, ":", " × "),
         Effect = str_replace_all(Effect, "Freq partner physical", "Physical"),
         df = paste0(Df, ", ", round(DenDF, 2))) |>
  filter(Effect != "Residuals") |>
  rename("F" = "F value",
         "p" = "Pr(>F)") |>
  mutate(F = round(F, 2),
         p = pval.lev(p)) |>
  select(Effect, F, df, p) |>
  mutate(omega = omega$Omega2_partial) |>
  add_row(Effect = paste0("Unadjusted = ", signif(r2$R2, 2))) |>
  add_row(Effect = paste0("Adjusted = ", signif(r2$R2_adjusted, 2))) |>
  rename("Fixed effect" = "Effect",
         "$F$" = "F",
         "$df$" = "df",
         "$p$" = "p",
         "$\\omega^2_p$" = "omega")

caption <- paste0("ANOVA-type table of fixed effects for the ", find_response(model), " model")

n_rows <- dim(tab)[1]

taa <- kable(tab,
              digits = 2,
              booktabs = TRUE,
              align = c("l", rep("c", 3)),
              linesep = "",
              caption = caption,
              escape = FALSE) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  pack_rows("$R^2$",
            start_row = n_rows-1, end_row = n_rows,
            hline_after = TRUE, hline_before = TRUE,
            escape = FALSE)
  return(list(tab = tab, kab = taa))
}

```

#### 1.2.4 main.eff.contr

This function takes a model, and creates an table of estimated marginal means and contrast between responses to masculinized and feminized responses, formatted in L<sup>A</sup>T<sub>E</sub>X.

```

main.eff.contr <- function(model, emm_contr) {

  headers <- c(5, 4)
  names(headers) <- c("",
                      paste0("Contrasts (",

```

```

    first(data.frame(emm_contr$contrast)$contrast),
    ")"))

merge(data.frame(emm_contr$emmeans),
      data.frame(emm_contr$contrast) |>
        mutate(p.value = pval.lev(p.value)) |>
        add_row(),
      by = 0) |>
  select(-c(Row.names, df.x, df.y, contrast)) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 8)),
        linesep = "",
        caption = paste0("Estimated marginal and contrast between ",
                        str_replace_all(emm_contr$emmeans@roles$predictors[1], "_", " "),
                        " levels for the ",
                        find_response(model), " model"),
        col.names = c(str_replace_all(emm_contr$emmeans@roles$predictors[1], "_", " "),
                     "EMM",
                     "$SE$",
                     "$2.5\\%CI$",
                     "$97.5\\%CI$",
                     "Difference",
                     "$SE$",
                     "$z$",
                     "$p$"),
        escape = FALSE) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  add_header_above(header = headers) |>
  footnote(general = "EMM = estimated marginal mean. No degrees of freedom are reported,
as an asymptotic method was used. Because of this, \textit{z} rather than
\textit{t} scores are reported. Significant effects are in bold.",
threeparttable = TRUE,
footnote_as_chunk = TRUE,
escape = FALSE)
}

```

### 1.2.5 inter.contr

This function takes a model, and creates an table of estimated marginal means and contrast between responses to masculinized and feminized responses, formatted in L<sup>A</sup>T<sub>E</sub>X.

```

inter.contr <- function(model, emm_contr) {

  headers <- c(6, 4)
  names(headers) <- c(" ",
                      paste0("Contrasts (",
                            first(data.frame(emm_contr$contrast)$contrast),
                            ")"))

  contr_levs <- paste0("Contrasts (",
                       first(data.frame(emm_contr$contrast)$contrast),
                       ")")

  merge(data.frame(emm_contr$emmeans),
        data.frame(emm_contr$contrast) |>
          mutate(p.value = pval.lev(p.value)) |>

```

```

add_row(.after = 1) |> add_row(),
by = 0) |>
rename_with(~ str_remove_all(., ".x")) |>
select(3,2,4,5,7,8,11,12,14,15) |>
kable(digits = 2,
booktabs = TRUE,
align = c(rep("l",2), rep("c",8)),
linesep = "",
caption = paste0("Estimated marginal and contrast between ",
str_replace_all(emm_contr$emmeans@roles$predictors[1], "_", " "),
" levels by ",
emm_contr$emmeans@roles$predictors[2],
" for the ",
find_response(model), " model"),
col.names = c(emm_contr$emmeans@roles$predictors[2],
str_replace_all(emm_contr$emmeans@roles$predictors[1], "_", " "),
"EMM",
"$SE$",
"$2.5\\%CI$",
"$97.5\\%CI$",
"Difference",
"$SE$",
"$z$",
"$p$"),
escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
add_header_above(header = headers) |>
collapse_rows(1,
              latex_hline = "major",
              row_group_label_position = "first") |>
footnote(general = "EMM = estimated marginal mean. No degrees of freedom are reported,
as an asymptotic method was used. Because of this, \\\textit{z} rather than
\\\\textit{t} scores are reported. Significant effects are in bold.",
threeparttable = TRUE,
footnote_as_chunk = TRUE,
escape = FALSE)
}

```

### 1.2.6 full.contr

This function takes a model, and creates an table of estimated marginal means and contrast between responses to masculinized and feminized responses by another variable, formatted in L<sup>A</sup>T<sub>E</sub>X.

```

full.contr <- function(model, emm_contr) {

headers <- c(7, 4)
names(headers) <- c(" ",
paste0("Contrasts (",
first(data.frame(emm_contr$contrast)$contrast),
")))

merge(data.frame(emm_contr$emmeans),
data.frame(emm_contr$contrast) |>
mutate(p.value = pval.lev(p.value)) |>
add_row(.after = 1) |> add_row(.after = 3) |>
add_row(.after = 5) |> add_row() |>
mutate(contrast = "Masculinized - Feminized"),

```

```

    by = 0) |>
  rename_with(~ str_remove_all(., ".x")) |>
  select(4,3,2,5,6,8,9,13,14,16,17) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c(rep("l",3), rep("c",8)),
        linesep = "\n",
        caption = paste0("Estimated marginal and contrast between masculinized and feminized
                         stimuli by Relationship and Conditionfor the ",
                         find_response(model), " model"),
        col.names = c(emm_contr$emmeans@roles$predictors[3],
                     emm_contr$emmeans@roles$predictors[2],
                     "Sexual dimorphism",
                     "EMM",
                     "$SE$",
                     "$2.5\\%CI$",
                     "$97.5\\%CI$",
                     "Difference",
                     "$SE$",
                     "$z$",
                     "$p$"),
        escape = FALSE) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  add_header_above(header = headers) |>
  collapse_rows(c(1:2,8),
                latex_hline = "major",
                row_group_label_position = "first") |>
  footnote(general = "EMM = estimated marginal mean. No degrees of freedom are reported,
               as an asymptotic method was used. Because of this, \textit{z} rather than
               \textit{t} scores are reported. Significant effects are in bold.",
            threeparttable = TRUE,
            footnote_as_chunk = TRUE,
            escape = FALSE)
}

```

### 1.2.7 `mods.comp`

This function takes a group of comparable models, and creates an table *AICc* values, including deltas and Akaike weights, formatted in L<sup>A</sup>T<sub>E</sub>X.

```

mods.comp <- function(base.model) {

  model.names <- c(base.model, paste0(base.model, letters[1:5]))

  model.list <- lapply(model.names, get)

  comp <- AICctab(model.list,
                  base = TRUE,
                  weights = TRUE,
                  mnames = model.names)

  tab <- data.frame(AICctab(model.list,
                             base = TRUE,
                             weights = TRUE,
                             mnames = c("(No covariate)",
                                         "Men perceived as dangerous",
                                         "Physical violence",
                                         "Women perceived as dangerous",
                                         "Women perceived as dangerous and physical violence"))

```

```

    "Sexual violence",
    "Infidelity",
    "Perceived home safety))) |>
rownames_to_column(var = "Model (by covariate)") |>
mutate("$Prob.$" = paste0("1 / ", number(max(weight)/weight, big.mark = ",")),
      AICc = number(AICc, big.mark = ","),
      weight = small.lev(weight)) |>
rename("$AICc$" = "AICc",
      "$\\Delta(AICc)$" = "dAICc",
      "$w(AICc)$" = "weight") |>
select(-c(df))

caption <- paste0("AICc comparison of ",
                  find_response(model.list[[1]]),
                  " models with different covariates")

taa <- kable(tab,
             digits = 2,
             booktabs = TRUE,
             align = c("l", rep("c", 3), "r"),
             linesep = "",
             caption = caption,
             escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position")) |>
footnote(general = "Models are sorted according to their $AICc$ values, increasingly,
so that the best model (lowest $AICc$) is always at the top.
$\\Delta(AICc)$ is the $AICc$ difference from each model to the best (top) model.
$w(AICc)$, are Akaike weights, which represent the 'weight of evidence'
(\\cite{portetPrimerModelSelection2020}) of each model for being the best model
of the collection, given the data.
Similarly, \textit{Prob.} refers to the probability of each model of being the
best model, in relation to the best-supported (top) model, expressed as a
fraction. For the model with no covariates, \textit{df} = 13; for all models
with covariates, \textit{df} = 21.",
            threeparttable = TRUE,
            footnote_as_chunk = TRUE,
            escape = FALSE)

return(list(tab = tab, kab = taa, comp = comp))
}

```

### 1.2.8 plot.exp

This function takes a model, and creates a plot that shows the estimated marginal means.

```

cond_labs <- c("Condition: High", "Condition: Low")
names(cond_labs) <- c("High", "Low")

plot.exp <- function(model, y.pos, y.lab) {

  emms_mod <- as.data.frame(emmeans(model,
                                      ~ Sexual_dimorphism + Condition + Relationship))

  contr_mod <- as.data.frame(pairs(emmeans(model,
                                           ~ Sexual_dimorphism | Condition + Relationship))) |>
  separate(contrast, c("group1", "group2"), " - ") |>
  mutate(p.signif = stars.pval(p.value))
}

```

```

plo <- ggplot(emms_mod, aes(y = emmean, x = Sexual_dimorphism, color = Relationship)) +
  geom_errorbar(aes(ymin = emmean-SE,
                     ymax = emmean+SE,
                     group = Relationship),
                color = "black",
                width=.2,
                position = position_dodge(0.3)) +
  geom_point(position = position_dodge(0.3), size = 2) +
  geom_line(aes(group = Relationship),
            position = position_dodge(0.3)) +
  stat_pvalue_manual(contr_mod,
                     label = "p.signif",
                     y.position = y.pos,
                     color = "Relationship", hide.ns = TRUE,
                     position = position_dodge(),
                     tip.length = 0) +
  labs(y = y.lab,
       x = NULL) +
  facet_grid(~ Condition,
             labeller = labeller(Condition = cond_labs))

return(plo)
}

```

### 1.2.9 plot.best.mod

This function takes a model with a covariate, and creates a plot that shows the estimated marginal means by levels of the covariate.

```

plot.best.mod <- function(best_model, y.pos, y.lab) {

  ifelse(class(best_model) == "lmerModLmerTest" |
    class(best_model) == "glmerMod",
    covar_best_mod <- best_model@frame |>
      select(where(is.numeric)) |>
      select(-1),
    covar_best_mod <- best_model$model |>
      select(where(is.numeric)) |>
      select(-1))

  covar_name <- colnames(covar_best_mod) |>
    str_replace_all("_", " ") |>
    str_remove_all("Freq partner") |>
    str_to_sentence() |>
    str_trim()

  covar_best_mod_levels <- c(min(covar_best_mod[,1]),
                               max(covar_best_mod[,1]))

  covar_labs <- paste0(covar_name,
                        c("\nMin = ", "\nMax = "),
                        round(covar_best_mod_levels, 3))

  names(covar_labs) <- covar_best_mod_levels

  cond_labs <- c("Condition: High", "Condition: Low")
}

```

```

names(cond_labs) <- c("High", "Low")

emms_best_mod <- as.data.frame(
  emmeans(
    best_model,
    ~ Sexual_dimorphism + Condition + Relationship + Freq_partner_physical_violence,
    at = list(Freq_partner_physical_violence = covar_best_mod_levels))) |>
  mutate(Freq_partner_physical_violence = round(Freq_partner_physical_violence, 3))

contr_best_mod <- as.data.frame(
  pairs(
    emmeans(
      best_model,
      ~ Sexual_dimorphism | Condition + Relationship + Freq_partner_physical_violence,
      at = list(Freq_partner_physical_violence = covar_best_mod_levels))) |>
  separate(contrast, c("group1", "group2"), " - ") |>
  mutate(p.signif = stars.pval(p.value))

best_mod_plot <- ggplot(emms_best_mod,
                        aes(y = emmean, x = Sexual_dimorphism, color = Relationship)) +
  geom_errorbar(aes(ymin = emmean-SE,
                     ymax = emmean+SE,
                     group = Relationship),
                color = "black",
                width=.1,
                position = position_dodge(0.3)) +
  geom_point(position = position_dodge(0.3), size = 1) +
  geom_line(aes(group = Relationship),
            position = position_dodge(0.3)) +
  stat_pvalue_manual(contr_best_mod, label = "p.signif",
                     y.position = y.pos,
                     color = "Relationship", hide.ns = TRUE,
                     position = position_dodge(),
                     tip.length = 0) +
  labs(y = y.lab,
       x = NULL) +
  facet_grid(Condition ~ Freq_partner_physical_violence,
             labeller = labeller(Condition = cond_labs,
                                  Freq_partner_physical_violence = covar_labs))
  return(best_mod_plot)
}

```

### 1.3 Independent stimuli evaluation

The sex typicality of all stimuli was manipulated to either enhance or reduce their sex-typical characteristics. Since all the stimuli were male faces, this involved masculinizing them to increase their typical sex characteristics and feminizing them to reduce those characteristics. Masculinized and feminized versions were independently rated for masculinity and estimated age by a panel of raters (not participants).

```
# Load the 'Evaluacion Manipulación Rostros.xlsx' Excel file into a data frame
ext_val <- read_excel("Data/Evaluacion Manipulación Rostros.xlsx")
```

#### 1.3.1 Masculinity ratings

First, masculinity rating given to the masculinized and feminized versions of each stimuli were compared.

```

# Select relevant columns and reshape the data into long format
masc_dat <- ext_val |>
  select(
    ResponseId,
    contains("M", ignore.case = FALSE), # Select columns that contain "M" (masculinity)
    -Menstruacion
  ) |> # Exclude the 'Menstruacion' (menstruation) column
  pivot_longer(
    cols = contains("M", ignore.case = FALSE), # Reshape to long format
    names_to = "Stimulus", # Column with stimuli names
    values_to = "Masculinity"
  ) |> # Column with masculinity ratings
  # Add a column indicating sexual dimorphism based on stimulus name
  mutate(Sexual_dimorphism = ifelse(grepl("f_1", Stimulus), "Feminine", "Masculine")) |>
  # Keep only the first 3 characters of the stimulus name
  mutate(Stimulus = str_sub(Stimulus, end = 3))

# Group by stimulus and perform t-tests for masculinity ratings across
# sexual dimorphism categories
t_masc <- masc_dat |>
  group_by(Stimulus) |>
  summarise(
    t = round(t.test(Masculinity ~ Sexual_dimorphism)$statistic, 2), # Compute t values
    p = pval.lev(t.test(Masculinity ~ Sexual_dimorphism)$p.value)
  ) |> # Compute p-value
  ungroup()

# Select the first 10 rows of the data 't_masc'
t_masc[1:10, ] |>
  # Add the next 10 rows (11 to 20) as additional columns
  cbind(t_masc[11:20, ]) |>
  # Add the next 10 rows (21 to 30) as additional columns
  cbind(t_masc[21:30, ]) |>
  # Create a table using the 'kable' function
  kable(
    booktabs = TRUE, # Use 'booktabs' style for better-looking tables in LaTeX
    digits = 2, # Round numerical values to 2 decimal places
    align = "c", # Center align all columns
    linesep = "", # No lines between rows
    caption = "Difference in independent masculinity ratings given to each stimulus,
              according to its sexual dimorphism manipulation (feminized - masculinized)",
    # Caption for the table
    escape = FALSE, # Allow LaTeX commands in the table (e.g., italic or bold)
    col.names = rep(c("Stimulus", "$t$"), times = 3) # Column names
  ) |>
  # Apply additional LaTeX styling to the table using 'kable_styling'
  kable_styling(
    latex_options = c("HOLD_position", "scale_down") # Keep table position
  ) |>
  # Add vertical lines after the 3rd and 6th columns using 'column_spec'
  column_spec(c(3, 6), border_right = TRUE) |>
  # Add a footnote with specific formatting
  footnote(
    general = "Tests are Welch's \\\textit{t}-test. Significant effects are in bold.",
    # General footnote text with LaTeX formatting
    threeparttable = TRUE, # Enable three-part table for better layout
  )

```

```

footnote_as_chunk = TRUE, # Render footnote as a chunk
escape = FALSE # Allow LaTeX commands in the footnote
)

```

**Table S1.** Difference in independent masculinity ratings given to each stimulus, according to its sexual dimorphism manipulation (feminized - masculinized)

Stimulus	t	p	Stimulus	t	p	Stimulus	t	p
A01	-6.09	< <b>0.0001</b>	A11	-6.98	< <b>0.0001</b>	A21	-7.81	< <b>0.0001</b>
A02	-9.05	< <b>0.0001</b>	A12	-7.90	< <b>0.0001</b>	A22	-10.53	< <b>0.0001</b>
A03	-8.96	< <b>0.0001</b>	A13	-10.32	< <b>0.0001</b>	A23	-6.83	< <b>0.0001</b>
A04	-8.04	< <b>0.0001</b>	A14	-7.76	< <b>0.0001</b>	A24	-6.61	< <b>0.0001</b>
A05	-9.81	< <b>0.0001</b>	A15	-10.33	< <b>0.0001</b>	A25	-8.18	< <b>0.0001</b>
A06	-7.45	< <b>0.0001</b>	A16	-10.63	< <b>0.0001</b>	A26	-8.60	< <b>0.0001</b>
A07	-7.04	< <b>0.0001</b>	A17	-7.76	< <b>0.0001</b>	A27	-6.55	< <b>0.0001</b>
A08	-9.05	< <b>0.0001</b>	A18	-10.29	< <b>0.0001</b>	A28	-7.79	< <b>0.0001</b>
A09	-12.18	< <b>0.0001</b>	A19	-8.27	< <b>0.0001</b>	A29	-11.25	< <b>0.0001</b>
A10	-6.53	< <b>0.0001</b>	A20	-9.72	< <b>0.0001</b>	A30	-11.47	< <b>0.0001</b>

*Note:* Tests are Welch's t-test. Significant effects are in bold.

### 1.3.2 Age ratings

Then, estimated age of stimuli was assessed.

```

# Process age-related data: select relevant columns and reshape into long format
age_dat <- ext_val |>
  select(
    ResponseId,
    contains("E", ignore.case = FALSE)
  ) |> # Select columns related to estimated age (E)
  select(-c(2:5)) |> # Exclude columns 2 to 5 (irrelevant for this analysis)
  pivot_longer(
    cols = contains("E", ignore.case = FALSE), # Reshape to long format
    names_to = "Stimulus", # Column with stimuli names
    values_to = "Age"
  ) |> # Column with age estimates
  # Add sexual dimorphism category based on stimulus name
  mutate(Sexual_dimorphism = ifelse(grepl("f_1", Stimulus), "Feminine", "Masculine")) |>
  # Keep only the first 3 characters of the stimulus name
  mutate(Stimulus = str_sub(Stimulus, end = 3))

# Summarize age data: compute mean, standard deviation, minimum, and maximum age
sum_age_dat <- age_dat |>
  summarise(
    Mean = mean(Age, na.rm = TRUE), # Mean age
    SD = sd(Age, na.rm = TRUE), # Standard deviation of age
    Min = min(Age, na.rm = TRUE), # Minimum age
    Max = max(Age, na.rm = TRUE)
  ) # Maximum age

```

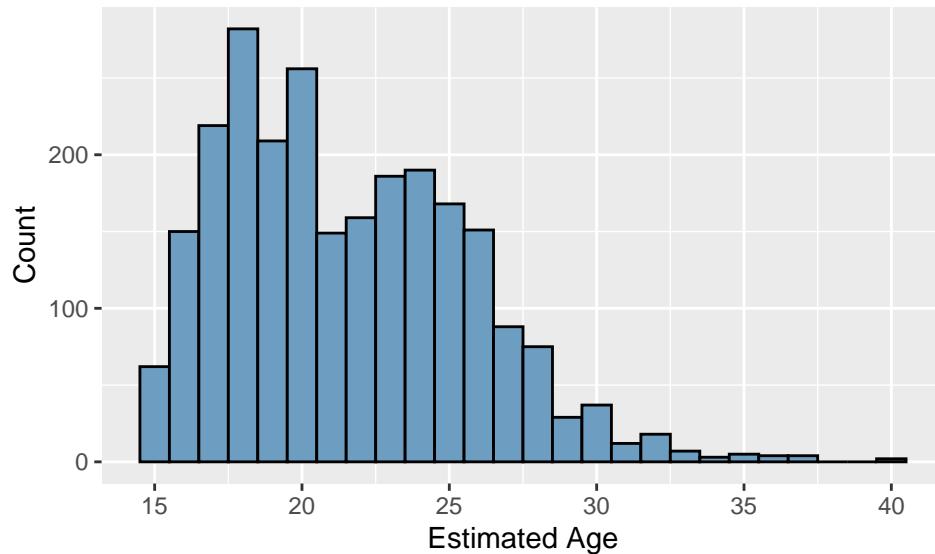
#### 1.3.2.1 Histogram of perceived age Distribution of the estimated ages.

```

# Create a histogram of estimated age
ggplot(age_dat, aes(x = Age)) +
  geom_histogram(bins = 26, fill = "#6D9EC1", color = "black") + # Plot histogram with 26 bins
  labs(

```

```
x = "Estimated Age", # X-axis label
y = "Count"
) +
scale_x_continuous(breaks = seq(15, 40, 5)) # X-axis scale with breaks every 5 units
```



**Figure S1.** Histogram of estimated age of stimuli by an independent panel of raters. Age estimations were between 15 and 40 with a mean of  $21.53 \pm 4.11$ .

## 1.4 Load and wrangle main experiment data

### 1.4.1 Individual databases (by data type/source)

```
# Load the 'CUC-UB' sheet from the 'BD-ET-CUC-UB.xlsx' dataset
dat_et <- read_excel("Data/BD-ET-CUC-UB.xlsx",
                      sheet = "CUC-UB")
) |>
# Drop unused columns
select(-c(
  Participant, Condicion, TOI, Interval, Media_respuesta, AOI,
  AOI_Global, Respuesta, Number_of_mouse_clicks...17,
  Time_to_first_mouse_click...18, AOI_respuesta
)) |>
# Rename columns (to English)
rename(
  ID = Recording,
  University = UNIVERSIDAD,
  Stimulus = Media,
  Condition = Condición,
  Relationship = Contexto,
  Sexual_dimorphism = Rostro,
  TFD = Total_duration_of_whole_fixations,
  NF = Number_of_whole_fixations,
  TFF = Time_to_first_whole_fixation,
  NMC = Number_of_mouse_clicks...21,
  TFMC = Time_to_first_mouse_click...22,
  DFF = Duration_first_fixation
) |>
```

```

# Convert character columns to factors
mutate(across(where(is.character), as.factor)) |>
# Recode factor levels to more meaningful English labels
mutate(
  Condition = fct_recode(Condition,
    "Low" = "BAJA",
    "High" = "ALTA"
  ),
  Relationship = fct_recode(Relationship,
    "Short term" = "CP",
    "Long term" = "LP"
  ),
  Sexual_dimorphism = fct_recode(Sexual_dimorphism,
    "Feminized" = "Feminizado",
    "Masculinized" = "Masculinizado"
  )
) |>
# Modify 'Stimulus' column to include 'F' for Feminized and 'M' for Masculinized
mutate(
  Stimulus = ifelse(Sexual_dimorphism == "Feminized",
    paste0(str_sub(str_replace(Stimulus, ".* - ", ""), 1, 2), "F"),
    ifelse(Sexual_dimorphism == "Masculinized",
      paste0(str_sub(str_replace(Stimulus, ".* - ", ""), 1, 2), "M"),
      Stimulus
    )
  ),
  # Create a new column 'CF' to indicate whether there was a mouse click
  CF = ifelse(NMC == 0, "No", "Yes")
)

```

#### 1.4.1.1 Eye-tracking data

**1.4.1.2 Questionnaires** This was loaded without calculating total instrument scores (for now), to test internal consistency

```

quests <- read_excel("Data/Cuestionario Datos Sociodemográficos (Disponibilidad) (respuestas) (1).xlsx",
                      sheet = "Respuestas de formulario 1"
) |>
# Drop unnecessary columns (such as 'Invitado', 'Servicios ayuda', and 'Correos cierre')
select(-c(Invitado, `Servicios ayuda`, `Correos cierre`)) |>
# Rename columns for better readability
rename(
  Date = Fecha,
  Age = edad,
  City = Ciudad,
  Education = Escolaridad,
  Ethnicity = Etnia,
  Gender = Sexo,
  Sex = Genero,
  Sexual_orientation = OS,
  Relationship_current = "Pareja actual",
  Relationship_duration = DuracionR,
  Relationship_status = EstadoR,
  Partner_sex = SexoParejaActual,
  Partner_masculinity = Masculinidad_pareja,
  Partner_dominance = Dominancia_pareja,
  Partner_attractiveness = Atractivo_pareja,

```

```

Number_of_children = NumHijos,
Hormonal_contraception = "Anticonceptivos hormonales",
Contraceptive = Cual_anticonceptivo,
Last_menstruation = "Ultima menstruacion",
Currently_pregnant = "Embarazo actual",
Sexual_abuse = "Experiencia abuso sexual",
Comments = comentarios1,
Medical_history = "antecedentes medicos",
SP_happiness = "AP felicidad",
SP_financial_security = "AP seguridad economica",
SP_money_control = "AP control dinero",
SP_attractiveness = "AP atractivo",
SP_self_confidence = "AP autoconfianza",
SP_self_esteem = "AP autoestima",
SP_health = "AP salud",
Electricity = "SB electricidad",
Internet_access = "SB internet",
TV = "SB television",
Internet_use = "Fr acceso internet",
Hospital_access = "Acceso hospital",
Freq_illness = "Fr enfermedades",
Socioeconomic_level = "Estrato socioeconomico",
Neighborhood = "Barrio de residencia",
Perceived_neighborhood_safety = "Seguridad barrio",
Perceived_city_safety = "Seguridad ciudad",
Perceived_home_safety = "Seguridad hogar",
Perceived_country_safety = "Seguridad país",
Freq_robbery = "Fr de robos",
Men_perceived_as_danger_to_children = "Hombres peligrosos hijos",
Men_perceived_as_danger_to_partner = "Hombres peligrosos pareja",
Partner_physical_violence = "VP fisica",
Freq_partner_physical_violence = "Fr VP fisica",
Partner_sexual_violence = "VP sexual",
Freq_partner_sexual_violence = "Fr VP sexual",
Partner_infidelity = "Infidelidad",
Freq_partner_infidelity = "Fr infidelidad",
Victim_of_violence = "Victima de alguna violencia",
Violence_type = "Tipo violencia",
Victim_of_gender_violence = "Victima violencia género",
Victim_of_armed_conflict = "Victima conflicto armado",
Control_question_1 = "Sin leer",
Control_question_2 = "Broma"
) |>
# Recode the factor levels of several categorical variables
mutate(
  Education = factor(Education, levels = c(
    "Primaria",
    "Bachillerato",
    "Universitario",
    "Posgrado"
  )),
  Sexual_orientation = factor(Sexual_orientation,
                               levels = c(
                                 "Exclusivamente heterosexual",
                                 "Principalmente heterosexual, con contactos homosexuales esporádicos",
                                 "Predominantemente heterosexual, aunque con contactos homosexuales más q

```

```

        "Bisexual",
        "Pansexual",
        "Demisexual"
    )
),
Relationship_status = factor(Relationship_status,
    levels = c(
        "Soltero sin contactos sexuales en el último año",
        "Soltero con contactos sexuales en el último año",
        "Relación exclusiva o matrimonio - viven juntos",
        "Relación exclusiva - no viven juntos",
        "Relación no exclusiva - contactos sexuales con otras personas"
    )
),
Internet_use = factor(Internet_use,
    levels = c("Cada día", "Cada mes", "Cada año")
),
Socioeconomic_level = as.factor(Socioeconomic_level)
) |>
# Recode City variable to simplify geographical information
mutate(City = ifelse(City %in% c(
    "Bogotá D.C.", "Madrid, Cundinamarca", "Zipaquirá, Cundinamarca",
    "Zipaquirá", "Mosquera, cundinamarca", "Mosquera",
    "FUNZA, CUNDINAMARCA", "Madrid Cundinamarca", "Une- Cundinamarca"
),
"Bogota Region",
ifelse(City %in% c(
    "Soledad", "Barranquilla", "BARRANQUILLA",
    "Soledad, Atlantico", "Costa Atlantica", "Corozal"
),
"Atlantico Region",
"Other"
)
)) |>
# Recode several factors from Spanish to English for easier interpretation
mutate(Education = recode(Education,
    "Primaria" = "Primary school",
    "Bachillerato" = "High school",
    "Universitario" = "University",
    "Posgrado" = "Postgraduate"
)) |>
# Additional recoding of variables
mutate(Sexual_orientation = recode(Sexual_orientation,
    "Exclusivamente heterosexual" =
        "Exclusively heterosexual",
    "Principalmente heterosexual, con contactos homosexuales esporádicos" =
        "Predominantly heterosexual",
    "Predominantemente heterosexual, aunque con contactos homosexuales más" =
        "Predominantly heterosexual, but more than incidentally homosexual",
    "Bisexual" = "Bisexual",
    "Pansexual" = "Pansexual",
    "Demisexual" = "Demisexual"
)) |>
mutate(Relationship_status = recode(Relationship_status,
    "Soltero sin contactos sexuales en el último año" =
        "Single without sexual contacts",

```

```

    "Soltero con contactos sexuales en el último año" =
    "Single with sexual contacts",
    "Relación exclusiva o matrimonio - viven juntos" =
    "Exclusive relationship - cohabitating",
    "Relación exclusiva - no viven juntos" =
    "Exclusive relationship - not cohabitating",
    "Relación no exclusiva - contactos sexuales con otras personas" =
    "Non-exclusive relationship"
)) |>
mutate(Internet_use = recode(Internet_use,
                               "Cada día" = "Daily",
                               "Cada mes" = "Monthly",
                               "Cada año" = "Yearly")
)) |>
# Recode several questions related to danger perceptions, replacing Spanish responses with
# numerical values.
mutate(across(
  starts_with("Men_perceived_as_danger_to_"),
  ~ recode(.,
            "Completamente en desacuerdo" = 1,
            "Ligeramente en desacuerdo" = 2,
            "Ni de acuerdo ni en desacuerdo" = 3,
            "Ligeramente deacuerdo" = 4,
            "Completamente deacuerdo" = 5
  )
)) |>
# Replace Spanish responses with corresponding English values
mutate(across(where(is.character), ~ replace(
  ., . ==
  "Si",
  "Yes"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . ==
  "Sí",
  "Yes"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . ==
  "No quiero responder",
  "Prefer not to answer"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . ==
  "Mujer",
  "Woman"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . ==
  "Hombre",
  "Man"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . ==
  "Femenino",
  "Female"

```

```

))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Masculino",
  "Male"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Sin pareja actual",
  "Single"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Sí, una vez en la adultez",
  "Once as adult"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Sí, tanto en la infancia como en la adultez",
  "Both as child and adult"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Sí, más de una vez en mi infancia",
  "More than once as child"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Sí, una vez e mi infancia",
  "Once as child"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Afrocolombiano",
  "Afrocolombian"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Desplazado conflicto armado",
  "Undetermined"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Ninguna",
  "Undetermined"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Comunidad negra",
  "Afrocolombian"
))) |>
mutate(across(where(is.character), ~ replace(
., . ==
  "Raizal del Archipiélago de San Andrés, Providencia y Santa Catalina",
  "Raizal"
))) |>

```

```

mutate(across(where(is.character), ~ replace(
  ., . == 
  "Patos",
  "Indigenous"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . == 
  "Indígena",
  "Indigenous"
))) |>
mutate(across(where(is.character), ~ replace(
  ., . == 
  "No estoy segura",
  "Unsure"
)))

```

**1.4.1.2.1 Principal component analysis (PCA)** To test whether it was possible to reduce the number of socio-ecological variables, we performed PCAs using the package FactoMineR (Lê et al., 2008), and plotted its results with function from the package factoextra (Kassambara & Mundt, 2020).

Socio-ecological factors PCA

```

# Select relevant columns for PCA from the 'quests' dataset
quests_pca_gen <- quests |>
  select(
    ID, # Unique identifier
    Men_perceived_as_danger_to_partner,
    Men_perceived_as_danger_to_children,
    Freq_partner_physical_violence,
    Freq_partner_sexual_violence,
    Freq_partner_infidelity,
    Perceived_home_safety
  ) |>
  # Rename columns: replace "Freq_" with "Frequency of"
  rename_with(~ str_replace_all(., "Freq_", "Frequency of")) |>
  # Replace underscores with spaces in column names
  rename_with(~ str_replace_all(., "_", " ")) |>
  # Capitalize the first letter of each column name
  rename_with(~ str_to_sentence(.))

# Perform PCA on the selected variables (excluding the ID column)
pca_sef <- PCA(quests_pca_gen[, -1], graph = FALSE)

# Display summary of the PCA results
pca_sef$var$cor |>
  # Create a table using the 'kable' function
  kable(
    booktabs = TRUE, # Use 'booktabs' style for better-looking tables in LaTeX
    digits = 2, # Round numerical values to 2 decimal places
    align = "c", # Center align all columns
    linesep = "", # No lines between rows
    caption = "Correlation between variables and PCA dimensions",
    # Caption for the table
    escape = FALSE, # Allow LaTeX commands in the table (e.g., italic or bold)
  ) |>
  # Apply additional LaTeX styling to the table using 'kable_styling'
  kable_styling(

```

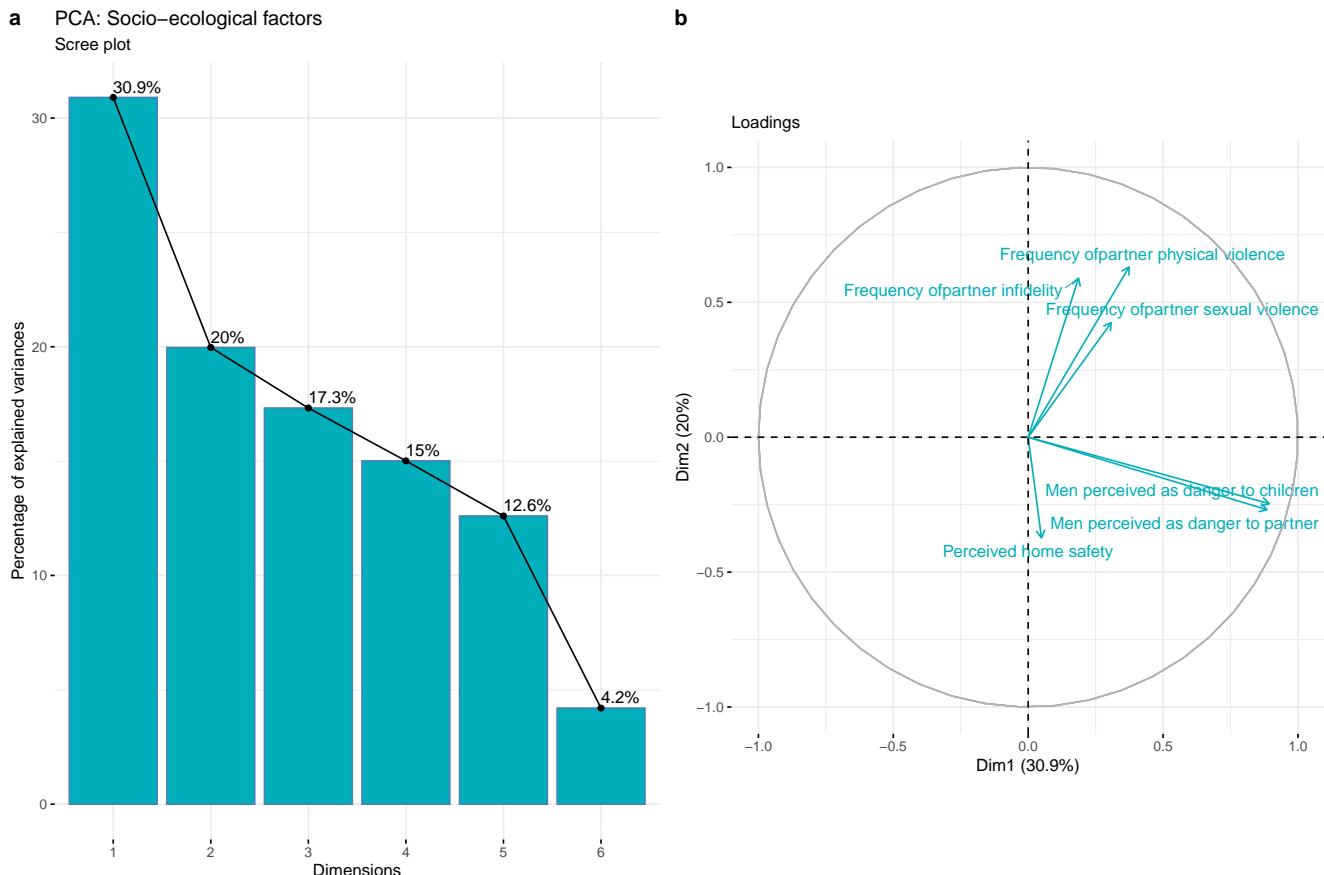
```
  latex_options = c("HOLD_position", "scale_down") # Keep table position
)
```

**Table S2.** Correlation between variables and PCA dimensions

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
Men perceived as danger to partner	0.89	-0.27	-0.03	-0.13	0.01
Men perceived as danger to children	0.89	-0.25	-0.04	-0.11	0.03
Frequency of partner physical violence	0.37	0.63	0.20	0.15	-0.63
Frequency of partner sexual violence	0.31	0.42	-0.48	0.61	0.35
Frequency of partner infidelity	0.19	0.59	0.54	-0.32	0.48
Perceived home safety	0.05	-0.37	0.69	0.61	0.06

### Summary plot

```
# Arrange two plots side by side:
# 1. A scree plot showing the explained variance for each principal component
# 2. A plot showing the variable loadings on the principal components
ggarrange(
  fviz_eig(pca_sef, addlabels = TRUE, barfill = "#00AFBB") +
    labs(
      title = "PCA: Socio-ecological factors", # Title for the scree plot
      subtitle = "Scree plot" # Subtitle for the scree plot
    ),
  fviz_pca_var(pca_sef,
    col.var = "#00AFBB", # Color the variable loadings in teal
    repel = TRUE # Avoid overlapping labels
  ) +
    labs(
      title = NULL, # No title for the loading plot
      subtitle = "Loadings" # Subtitle for the loading plot
    ),
  labels = "auto"
)
```



**Figure S2.** Summary of the PCA for all socio-ecological factors. **a.** Scree plot. **b.** Factor loadings.

When including all socio-ecological factors, the only variables that strongly correlate between them and with the PCA dimension (Table ??; Fig. S2), are the two variables that evaluate participant's perception of men as dangerous to children and to their partner.

Because of this, a new PCA was performed on only these two variables, to calculate a score of Men perceived as dangerous. All remaining socio-ecological variables were kept.

Men perceived as dangerous

```
# Select relevant columns for PCA from the 'quests' dataset
quests_pca <- quests |>
  select(
    ID,
    Men_perceived_as_danger_to_partner,
    Men_perceived_as_danger_to_children
  ) |>
  # Rename columns: remove "Men_perceived_as_danger_to_"
  rename_with(~ str_remove_all(., "Men_perceived_as_danger_to_")) |>
  # Capitalize the first letter of each column name
  rename_with(~ str_to_sentence(.))

# Perform PCA on the selected variables (excluding the ID column)
pca_mpd <- PCA(quests_pca[, -1], graph = FALSE)

# Calculate score for the men perceived as dangerous dimension
mpd_scores <- data.frame(pca_mpd$ind$coord)$Dim.1
```

```
# Display summary of the PCA results
pca_mpd$var$cor |>
  # Create a table using the 'kable' function
  kable(
    booktabs = TRUE, # Use 'booktabs' style for better-looking tables in LaTeX
    digits = 2, # Round numerical values to 2 decimal places
    align = "c", # Center align all columns
    linesep = "", # No lines between rows
    caption = "Correlation between variables and PCA dimensions",
    # Caption for the table
    escape = FALSE, # Allow LaTeX commands in the table (e.g., italic or bold)
  ) |>
  # Apply additional LaTeX styling to the table using 'kable_styling'
  kable_styling(
    latex_options = c("HOLD_position", "scale_down") # Keep table position
  )
```

**Table S3.** Correlation between variables and PCA dimensions

	Dim.1	Dim.2
Partner	0.93	0.36
Children	0.93	-0.36

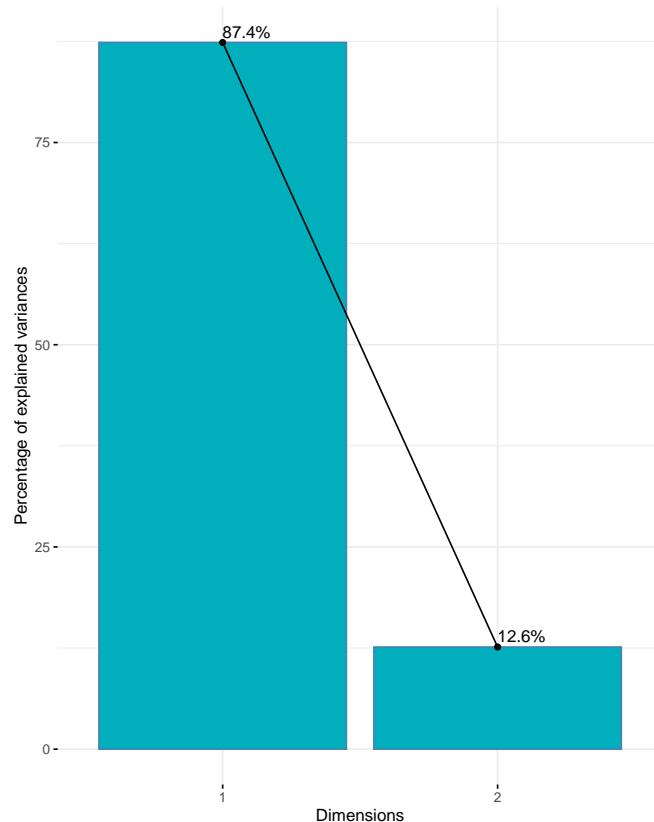
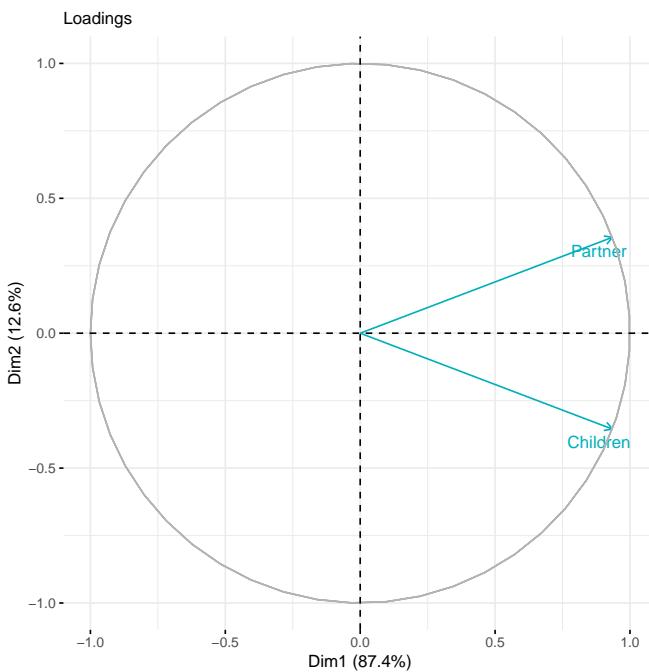
### Summary plot

In fact, the two variables related to men perceived as dangerous, could be reduced to a single dimension, that captured over 87% of the variance.

```
# Arrange two plots side by side:
# 1. A scree plot showing the explained variance for each principal component
# 2. A plot showing the variable loadings on the principal components
ggarrange(
  fviz_eig(pca_mpd, addlabels = TRUE, barfill = "#00AFBB") +
  labs(
    title = "PCA: Men perceived as danger to...", # Title for the scree plot
    subtitle = "Scree plot" # Subtitle for the scree plot
  ),
  fviz_pca_var(pca_mpd,
    col.var = "#00AFBB", # Color the variable loadings in teal
    repel = TRUE # Avoid overlapping labels
  ) +
  labs(
    title = NULL, # No title for the loading plot
    subtitle = "Loadings" # Subtitle for the loading plot
  ),
  labels = "auto"
)
```

**a** PCA: Men perceived as danger to...

Scree plot

**b****Figure S3.** Summary of the PCA for factors related to men perceived as dangerous. **a.** Scree plot. **b.** Factor loadings.

#### 1.4.1.2.2 Clean questionnaire data Less columns, with total instrument scores

```
# Clean and modify the 'quests' dataset
quests_clean <- quests |>
  # Recode values in columns that start with "Escasez alimentaria"
  mutate(across(
    starts_with("Escasez alimentaria"),
    ~ recode(.,
      "Nunca" = 0, # Recode "Nunca" to 0
      "Rara vez/algunas veces" = 1, # Recode "Rara vez/algunas veces" to 1
      "Casi siempre" = 2 # Recode "Casi siempre" to 2
    )
  )) |>
  # Perform row-wise operations
  rowwise() |>
  # Create new variables by summing up specific columns
  mutate(
    # Calculate Self-esteem score by summing relevant items (with reverse scoring)
    Self_esteem = sum(
      autoestima_I1, 5 - autoestima_I2, autoestima_I3, autoestima_I4,
      autoestima_I5, 5 - autoestima_I6, autoestima_I7, 5 - autoestima_I8,
      5 - autoestima_I9, autoestima_I10
    ),
    # Calculate Self-perception score by summing columns that start with "SP_"
    Self_perception = sum(across(starts_with("SP_")))
  )
```

```

# Calculate Perceived safety by summing columns that end with "_safety"
Perceived_safety = sum(across(ends_with("_safety"))),
# Calculate Food insecurity by summing columns that start with "Escasez alimentaria"
Food_insecurity = sum(across(starts_with("Escasez alimentaria")))
) |>
# Remove columns that start with "autoestima_"
select(!starts_with("autoestima_")) |>
# Convert character columns to factors
mutate(across(where(is.character), as.factor)) |>
# Bind the column 'Men_perceived_as_dangerous' from 'mpd_scores' (PCA scores)
bind_cols(Men_perceived_as_dangerous = mpd_scores)

```

#### 1.4.1.3 Subjective evaluation of stimuli

```

# Load the subjective evaluation dataset, removing the last two columns (123 and 124)
eval <- read_excel("Data/Evaluación subjetiva rostros (Respuestas).xlsx") |>
  select(-c(123:124)) |>
# Perform row-wise operations to compute new variables
  rowwise() |>
# Calculate the sum of masculinity and attractiveness ratings for both masculinized and
# feminized stimuli
  mutate(
    Masculinity_masculinized = sum(across(ends_with("M Mas"))),
    Masculinity_feminized = sum(across(ends_with("F Mas"))),
    Attractiveness_masculinized = sum(across(ends_with("M Atr"))),
    Attractiveness_feminized = sum(across(ends_with("F Atr"))))
  ) |>
# Rename columns for clarity
  rename(
    Date = "Marca temporal",
    ID = "Escribe tu código de participante"
  )

```

##### 1.4.1.3.1 Wide format

```

# Create a long format dataset by combining attractiveness and masculinity ratings
eval_long <- left_join(
# First, select relevant columns and pivot the attractiveness ratings to long format
  eval |>
    select(-c(123:126)) |> # Remove unnecessary columns
    select(!ends_with(" Mas")) |> # Exclude masculinity-related columns
    pivot_longer(
      cols = ends_with("Atr"), # Pivot attractiveness ratings to long format
      names_to = "Stimulus",
      values_to = "Attractiveness"
    ) |>
    mutate(Stimulus = str_remove_all(Stimulus, " Atr")), # Clean the Stimulus names
# Next, pivot the masculinity ratings to long format
  eval |>
    select(-c(123:126)) |> # Remove unnecessary columns
    select(!ends_with(" Atr")) |> # Exclude attractiveness-related columns
    pivot_longer(
      cols = ends_with("Mas"), # Pivot masculinity ratings to long format
      names_to = "Stimulus",

```

```

    values_to = "Masculinity"
) |>
  mutate(Stimulus = str_remove_all(Stimulus, " Mas")) # Clean the Stimulus names
)

```

#### 1.4.1.3.2 Long format

```

reg <- rbind(
  read_excel("Data/3Registro Participantes Disponibilidad de Recursos-corregido.xlsx",
             sheet = "UB"
  ) |>
    mutate(University = "UB"),
  read_excel("Data/3Registro Participantes Disponibilidad de Recursos-corregido.xlsx",
             sheet = "CUC"
  ) |>
    mutate(University = "CUC")
) |>
  select(-c(
    Grupo, `Entrega de kit`, `Protocolo de bioseguridad`, `Requisitos previos al registro`,
    Consentimiento, `Código de evaluador`:`Código auxiliar que reclutó`
  )) |>
  rename(
    Date = "Fecha de registro",
    ID = "Codigo del Participante",
    Condition = "Condicion",
    Calibration = "Calibración",
    Gaze_perc = "% Gaze",
    Condition_happiness = "Q Feliz",
    Condition_physical_safety = "Q Segura físicamente",
    Condition_healthy = "Q Saludable",
    Condition_economic_security = "Q Segura económicamente",
    Body_temperature = "Temperatura",
    Ovulating = "Test de ovulación",
    Saliva_pre = "Recolección de saliva pre",
    Saliva_pre_time = "Hora...18",
    Eye_tracking = "Rastreo Ocular",
    Subjective_evaluation = "Evaluación subjetiva",
    Sociodemographic_questionnaire = "Cuestionario sociodemográfico",
    Saliva_post = "Recolección de saliva post",
    Saliva_post_time = "Hora...23",
    Notes = "Observaciones"
  ) |>
  mutate(
    Condition = fct_recode(Condition,
                           "Low" = "Baja",
                           "High" = "Alta"
    ),
    Calibration = fct_recode(Calibration,
                             "<=0.5" = "<0.5 (menor a 0.5)",
                             ">0.5" = ">0.5 (mayor a 0.5)",
                             "<=0.5" = "0.5 (igual a 0.5)",
                             NULL = "Selecciona"
    ),
    Ovulating = fct_recode(as.factor(Ovulating),
                           "No" = "0",
                           "Yes" = "1"
    )
)

```

```

        "Yes" = "1"
    )
) |>
mutate_all(~ str_replace_all(., "SI", "Yes")) |>
mutate_all(~ str_replace_all(., "NO", "No")) |>
mutate_all(~ str_replace_all(., "INCOMPLETO", "No")) |>
mutate_all(~ str_replace_all(., "Recuperado", "Data recovered")) |>
mutate_all(~ str_replace_all(., "RECUPERADO", "Data recovered")) |>
mutate_all(~ na_if(., "Selecciona")) |>
mutate_all(~ na_if(., "N/A")) |>
mutate(across(starts_with("Condition_"), as.numeric))

```

#### 1.4.1.4 Resource availability

#### 1.4.2 Full, final database

```

dat_int <- dat_et |>
left_join(quests_clean, by = c("ID"), multiple = "all") |>
left_join(eval_long, by = c("ID", "Stimulus"), multiple = "all") |>
left_join(reg, by = c("ID", "University", "Condition"), multiple = "all")

```

#### 1.4.2.1 Join data files

**1.4.2.2 Filtered database** Filtered database to exclude participants who did responded the two control questions correctly, were ovulating, or did not report being exclusively heterosexual.

```

dat <- dat_int |>
# Filter out rows where Control_question_1 and Control_question_2 are both "No",
# Ovulating is not "Yes", and Sexual_orientation is "Exclusively heterosexual"
filter(Control_question_1 == "No" &
      Control_question_2 == "No" &
      Ovulating != "Yes" &
      Sexual_orientation == "Exclusively heterosexual") |>
# Remove all occurrences of the letter "F" from the Stimulus column
# (infomation already in the column Sexual_dimorphism)
mutate(Stimulus = str_remove_all(Stimulus, "F")) |>
# Remove all occurrences of the letter "M" from the Stimulus column
# (infomation already in the column Sexual_dimorphism)
mutate(Stimulus = str_remove_all(Stimulus, "M")) |>
# Sort Sexual_dimorphism levels, so that contrasts keep the same structure
mutate(Sexual_dimorphism = fct_relevel(Sexual_dimorphism, "Masculinized")) |>
# Ensure that the resulting data frame is ungrouped
ungroup()

```

After filtering the database and removing data who did not meet these criteria, from an initial sample size of 499 women, the final database contained data from 293 exclusively heterosexual participants, who were not ovulating.

#### 1.4.3 Final individual databases filtered to the final sample

```

reg_fin <- reg |>
left_join(quests_clean, by = c("ID")) |>
filter(ID %in% unique(dat$ID))

```

#### 1.4.3.1 Resource availability (filtered)

```
quests_fin <- quests_clean |>
  filter(ID %in% unique(dat$ID))
```

#### 1.4.3.2 Questionnaires (filtered)

## 2 Descriptives

### 2.1 Number and age of participants in each condition

```
dat |>
  group_by(ID) |>
  summarise(
    Age = first(Age),
    Condition = first(Condition)
  ) |>
  ungroup() |>
  group_by(Condition) |>
  summarise(
    n = n_distinct(ID),
    Mean = mean(Age, na.rm = TRUE),
    SD = sd(Age, na.rm = TRUE),
    Min = min(Age, na.rm = TRUE),
    Max = max(Age, na.rm = TRUE)
  ) |>
  kable(
    booktabs = TRUE, # Use 'booktabs' style for better-looking tables in LaTeX
    digits = 2, # Round numerical values to 2 decimal places
    align = "c", # Center align all columns
    linesep = "", # No lines between rows
    caption = "Number and age of participants in each condition",
    # Caption for the table
    escape = FALSE, # Allow LaTeX commands in the table (e.g., italic or bold)
    col.names = c(
      "Condition",
      "\\textit{n}",
      "Mean",
      "SD",
      "Min.",
      "Max."
    )
  ) |>
  # Apply additional LaTeX styling to the table using 'kable_styling'
  kable_styling(
    latex_options = c("HOLD_position", "scale_down") # Keep table position
  )
```

**Table S4.** Number and age of participants in each condition

Condition	n	Mean	SD	Min.	Max.
High	165	21.41	2.25	18	27
Low	128	21.50	2.25	18	25

## 2.2 Select and wrangle data for descriptive plots

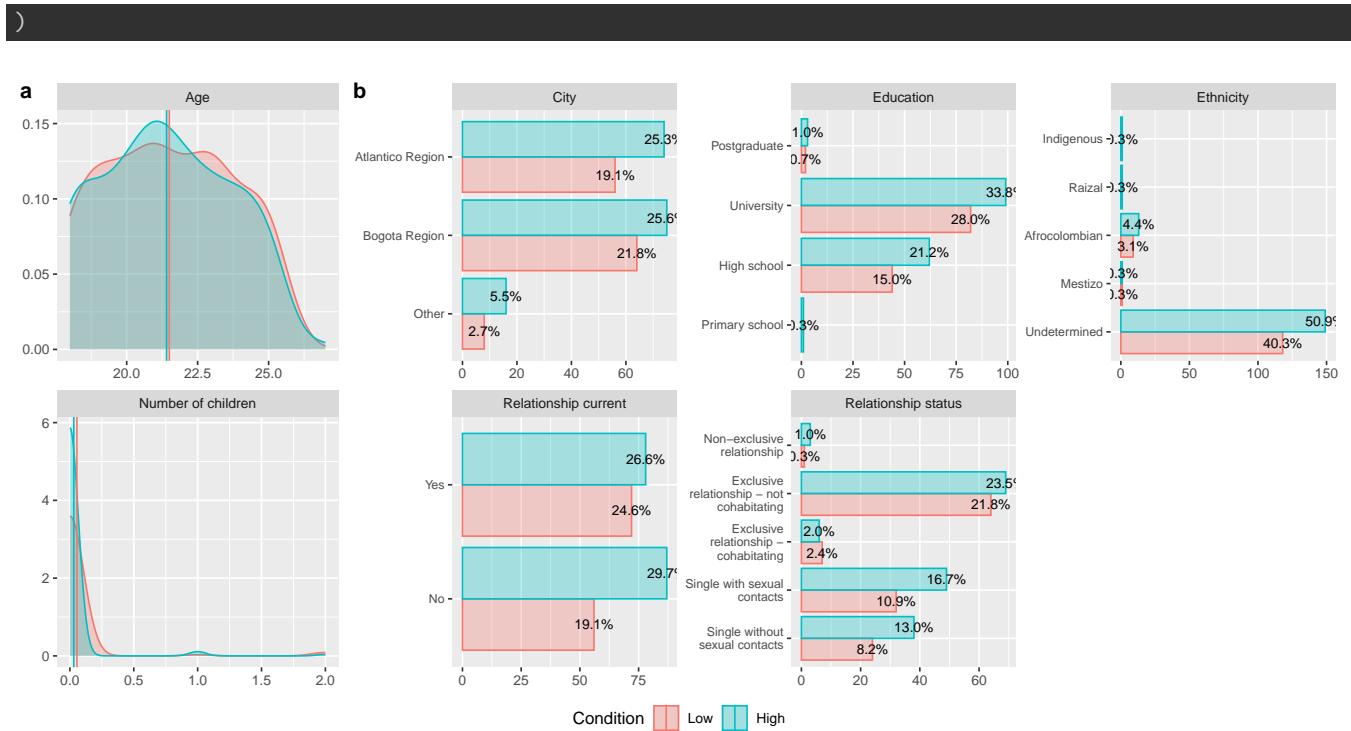
```
# Create desc_quest, combining and transforming quests_fin and reg
desc_quest <-
  # Join the quests_fin and reg dataframes by ID
  quests_fin |>
  left_join(reg, by = c("ID")) |>
  # Select only the desired columns
  select(
    ID,
    Condition,
    Age,
    City,
    Education,
    Ethnicity,
    Sexual_orientation,
    Relationship_current,
    Relationship_status:Hormonal_contraception,
    Sexual_abuse,
    SP_happiness:Socioeconomic_level,
    Perceived_country_safety:Freq_robbery,
    Victim_of_violence,
    Victim_of_gender_violence:Victim_of_armed_conflict,
    Self_esteeem:Men_perceived_as_dangerous,
    Freq_partner_physical_violence,
    Freq_partner_infidelity,
    Partner_physical_violence,
    Partner_sexual_violence,
    Freq_partner_sexual_violence,
    # Food security variables (transformed later)
    "Escasez alimentaria1":"Escasez alimentaria5"
  ) |>
  # Transform all 'Escasez alimentaria' (food scarcity) columns into categorical variables with
  # specific levels.
  mutate(
    across(
      starts_with("Escasez alimentaria"),
      ~ recode(.,
        "0" = "Never",
        "1" = "Rarely/sometimes",
        "2" = "Almost always"
      )
    ),
    # Convert character variables to factor for clarity and consistency.
    across(where(is.character), as.factor),
    # Sort factor levels
    across(
      starts_with("Escasez alimentaria"),
      ~ factor(.,
        levels = c(
          "Never",
          "Rarely/sometimes",
          "Almost always"
        )
      )
    )
  )
```

)

## 2.3 Distribution of values across variables

### 2.3.1 Sociodemographic variables

```
# Create a plot that displays the distribution of sociodemographic factors
# by condition, with subplots for numeric and categorical variables.
ggarrange(
  # Plot a: Distribution of values across numeric sociodemographic variables
  desc_quest |>
    select(ID, Condition, Age, Number_of_children) |>
    # Convert data from long to wide format to prepare for plotting
    pivot_longer(where(is.numeric),
      names_to = "Variable",
      values_to = "Value") |>
    # Clean and transform the variable names by replacing underscores with spaces
    mutate(Variable = str_replace_all(Variable, "_", " ")) |>
    # Create a plot of density distributions for numeric variables,
    # colored and filled by condition
    ggplot(aes(x = Value, fill = Condition, color = Condition)) +
    geom_density(alpha = 0.3) + # Use semi-transparent density curves
    facet_wrap(~Variable, scales = "free", ncol = 1) + # Display variables in separate panels
    stat_summary(aes(xintercept = after_stat(x), y = 0),
      fun = mean, geom = "vline", orientation = "y") + # Add vertical lines at mean
    labs(x = NULL, y = NULL), # Remove axis labels for this panel
  # Plot b: Proportional number of participants across categorical variables
  desc_quest |>
    select(ID, Condition, City, Ethnicity,
      Education, Relationship_current, Relationship_status) |>
    # Convert data from long to wide format to prepare for plotting
    pivot_longer(City:Relationship_status,
      names_to = "Variable",
      values_to = "Value") |>
    # Clean and transform the variable names by replacing underscores with spaces
    mutate(Variable = str_replace_all(Variable, "_", " ")) |>
    # Create a plot of bar charts for categorical variables,
    # colored and filled by condition
    ggplot(aes(y = Value, fill = Condition, color = Condition)) +
    geom_bar(alpha = 0.3, position = position_dodge()) + # Use semi-transparent bars
    # Add text labels to display proportional values as percentages
    geom_text(aes(label = scales::percent(after_stat(prop)), accuracy = 0.1)),
      xjust = "inward",
      position = position_dodge(.9),
      stat = "prop",
      color = "black",
      size = 3) +
    facet_wrap(~Variable, scales = "free") + # Display variables in separate panels
    scale_y_discrete(labels = label_wrap(20)) + # Wrap long labels for categorical axes
    theme(axis.text.y = element_text(size = 8)) + # Reduce font size for y-axis text
    labs(x = NULL, y = NULL), # Remove axis labels for this panel
  # Arrange subplots into a grid with specified widths and share legends
  widths = c(1, 3),
  common.legend = TRUE,
  legend = "bottom",
  labels = "auto"
```



**Figure S4.** Distribution of values across sociodemographic variables, by condition. **a.** Distribution of values across numeric sociodemographic variables. Colored vertical lines indicate the mean value for each variable under each condition. **b.** Proportional number of participants across categorical variables.

### 2.3.2 Access to resources

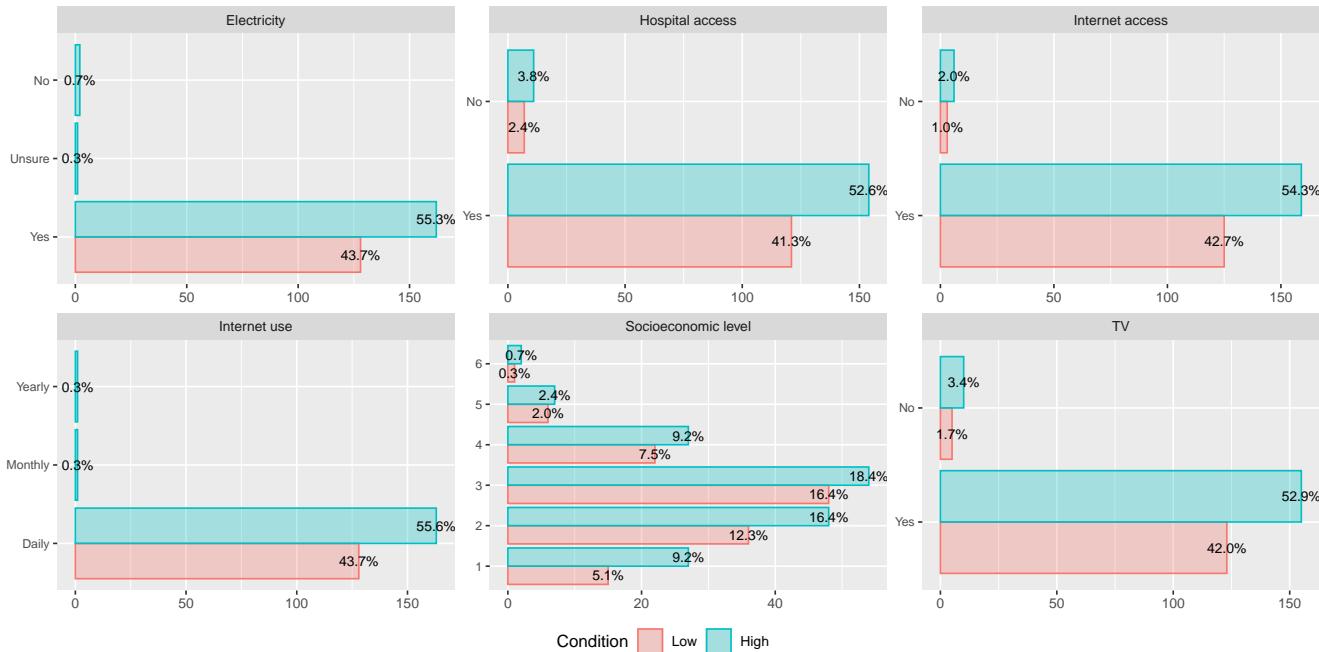
```
# Create a plot that displays the distribution of socioeconomic factors
# by condition.

ggarrange(
  # Select relevant variables from the dataset (desc_quest)
  desc_quest |>
    select(ID, Condition,
           Socioeconomic_level, Electricity, Internet_access, Internet_use,
           TV, Hospital_access) |>
  # Convert data from long to wide format to prepare for plotting
  pivot_longer(Socioeconomic_level:Hospital_access,
               names_to = "Variable",
               values_to = "Value") |>
  # Clean and transform the variable names by replacing underscores with spaces
  mutate(Variable = str_replace_all(Variable, "_", " ")) |>
  # Create a plot of bar charts for socioeconomic variables,
  # colored and filled by condition
  ggplot(aes(y = Value, fill = Condition, color = Condition)) +
  geom_bar(alpha = 0.3, position = position_dodge()) + # Use semi-transparent bars
  # Add text labels to display proportional values as percentages
  geom_text(aes(label = scales::percent(after_stat(prop), accuracy = 0.1)),
            xjust = "inward",
            position = position_dodge(.9),
            stat = "prop",
            color = "black",
            size = 3) +
  facet_wrap(~Variable, scales = "free") + # Display variables in separate panels
```

```

scale_y_discrete(labels = label_wrap(20)) + # Wrap long labels for categorical axes
theme(axis.text.y = element_text(size = 8)) + # Reduce font size for y-axis text
labs(x = NULL, y = NULL), # Remove axis labels
# Arrange subplots into a grid with specified widths and share legends
widths = c(1, 3),
common.legend = TRUE,
legend = "bottom"
)

```



**Figure S5.** Proportional number of participants across categorical variables that measure access to resources.

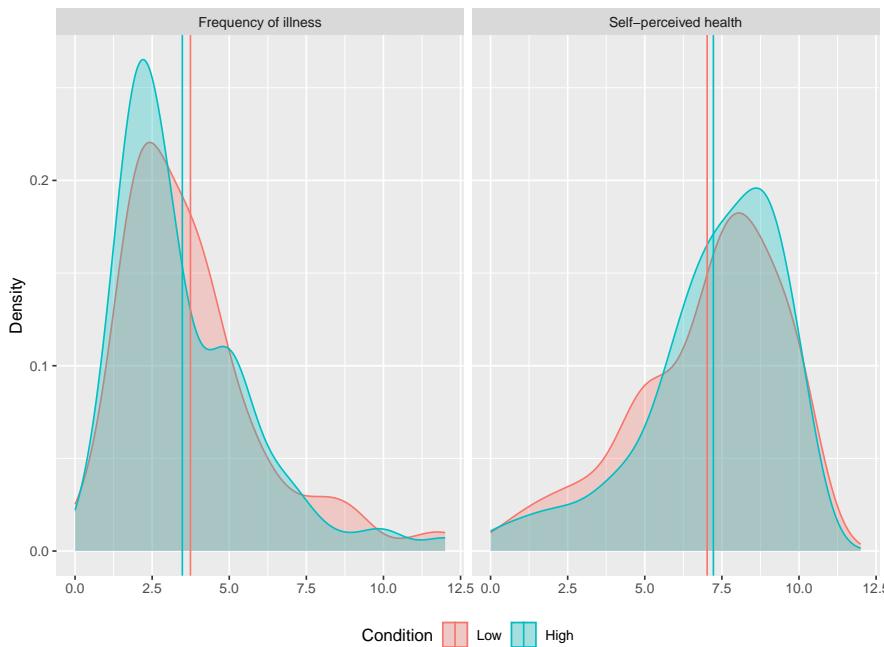
### 2.3.3 Health-related variables

```

ggarrange(
  # Select relevant columns from desc_quest and pivot them into a long format
  desc_quest |>
    select(ID, Condition, Freq_illness, SP_health) |>
    # Convert the Frequency of illness and Self-perceived health columns into separate rows
    pivot_longer(Freq_illness:SP_health,
      names_to = "Variable",
      values_to = "Value") |>
    # Clean up variable names by replacing underscores with spaces
    mutate(Variable = str_replace_all(Variable, "_", " ")) |>
    # Rename variables
    mutate(Variable = str_replace_all(Variable, "Freq", "Frequency of")) |>
    mutate(Variable = str_replace_all(Variable, "SP", "Self-perceived")) |>
    # Convert the Value column to numeric
    mutate(Value = as.numeric(Value)) |>
    # Create a ggplot object
    ggplot(aes(x = Value, fill = Condition, color = Condition)) +
    # Plot density curves for each condition within each variable
    geom_density(alpha = 0.3) +
    # Divide the plot into facets by Variable
    facet_wrap(~Variable) +

```

```
# Add vertical lines to indicate mean values for each group
stat_summary(aes(xintercept = after_stat(x), y = 0),
             fun = mean, geom = "vline", orientation = "y") +
# Set up plot labels and title with NULL values for x and y axes.
labs(x = NULL, y = "Density"),
# Specify the widths of the two columns and common legend position (bottom)
widths = c(2, 1),
common_legend = TRUE,
legend = "bottom")
```



**Figure S6.** Distribution of values across numeric health-related variables. Colored vertical lines indicate the mean value for each variable under each condition.

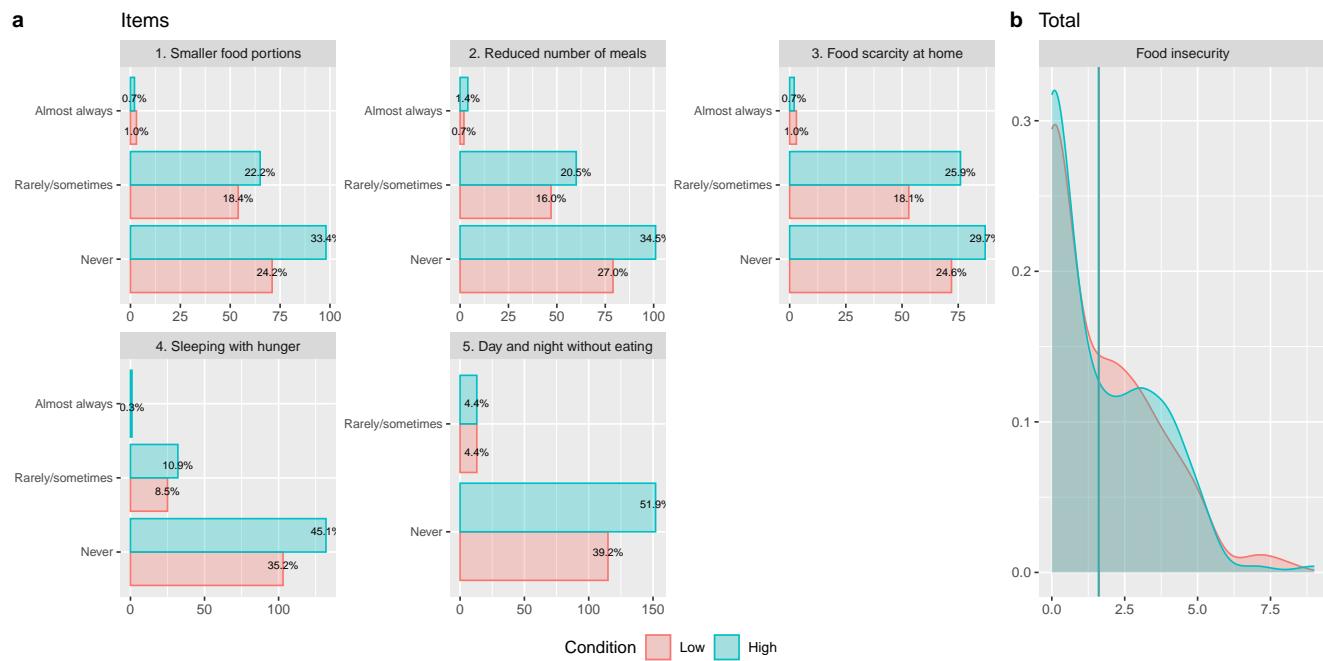
### 2.3.4 Food security

```
ggarrange(
  # Select columns from desc_quest, including 'Escasez alimentaria' (food scarcity)
  desc_quest |>
    select(ID, Condition, "Escasez alimentaria1":"Escasez alimentaria5") |>
    # Pivot the Escasez alimentaria1 to Escasez alimentaria5 columns into a long format
    pivot_longer("Escasez alimentaria1":"Escasez alimentaria5",
                 names_to = "Variable",
                 values_to = "Value") |>
    # Clean up variable names
    mutate(Variable = str_replace_all(Variable, "Escasez alimentaria", "")) |>
    mutate(Variable = str_replace_all(Variable, "1", "1. Smaller food portions")) |>
    mutate(Variable = str_replace_all(Variable, "2", "2. Reduced number of meals")) |>
    mutate(Variable = str_replace_all(Variable, "3", "3. Food scarcity at home")) |>
    mutate(Variable = str_replace_all(Variable, "4", "4. Sleeping with hunger")) |>
    mutate(Variable = str_replace_all(Variable, "5", "5. Day and night without eating")) |>
    # Create a ggplot object for the first set of data
    ggplot(aes(y = Value, fill = Condition, color = Condition)) +
    # Plot bar charts for each condition within each variable
    geom_bar(alpha = 0.3, position = position_dodge()) +
```

```

# Add text labels on top of the bars showing the proportion of each category
geom_text(aes(label = scales::percent(after_stat(prop), accuracy = 0.1)),
          vjust = "inward",
          position = position_dodge(.9),
          stat = "prop",
          color = "black",
          size = 2.5) +
# Divide the plot into facets by Variable
facet_wrap(~Variable, scales = "free") +
# Set labels for the y-axis with a maximum width of 20 characters
scale_y_discrete(labels = label_wrap(20)) +
# Adjust the text size of the y-axis
theme(axis.text.y = element_text(size = 8)) +
labs(x = NULL, y = NULL, title = "Items"),
# Select columns from desc_quest dataframe, including Food_insecurity column
desc_quest |>
  select(ID, Condition, Food_insecurity) |>
# Convert the Food_insecurity column into long format
pivot_longer(Food_insecurity,
             names_to = "Variable",
             values_to = "Value") |>
# Convert the Value column to numeric
mutate(Value = as.numeric(Value)) |>
# Clean up variable names
mutate(Variable = str_replace_all(Variable, "_", " ")) |>
# Create a ggplot object for the second set of data
ggplot(aes(x = Value, fill = Condition, color = Condition)) +
# Plot density curves for each condition within each variable
geom_density(alpha = 0.3) +
# Divide the plot into facets by Variable
facet_wrap(~Variable) +
# Add vertical lines to indicate mean values
stat_summary(aes(xintercept = after_stat(x), y = 0),
             fun = mean, geom = "vline", orientation = "y") +
labs(x = NULL, y = NULL, title = "Total"),
# Specify the widths of the two columns and common legend position (bottom)
widths = c(3, 1),
common.legend = TRUE,
legend = "bottom",
labels = "auto")

```

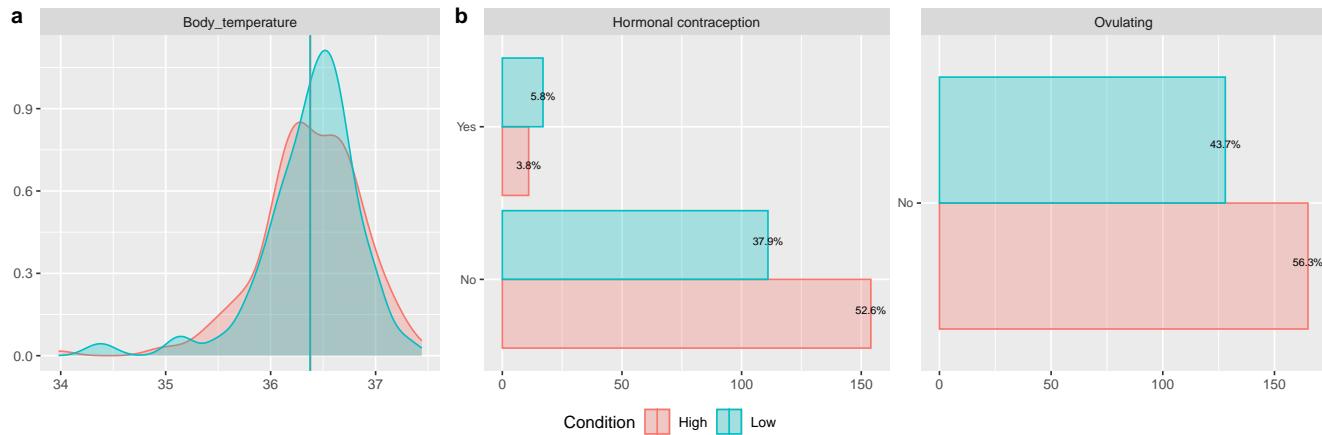


**Figure S7.** Distribution of values across food security variables, by condition. **a.** Proportional number of participants across ordinal items. **b.** Distribution of values for the total score. Colored vertical lines indicate the mean value for participants in each condition.

### 2.3.5 Hormonal variables

```
ggarrange(
  reg_fin |>
    select(ID, Condition, Body_temperature) |>
    pivot_longer(Body_temperature,
      names_to = "Variable",
      values_to = "Value") |>
    mutate(Value = as.numeric(Value)) |>
    ggplot(aes(x = Value, fill = Condition, color = Condition)) +
    geom_density(alpha = 0.3) +
    facet_wrap(~Variable) +
    stat_summary(aes(xintercept = after_stat(x), y = 0),
      fun = mean, geom = "vline", orientation = "y") +
    labs(x = NULL, y = NULL),
  reg_fin |>
    left_join(desc_quest, by = c("ID", "Condition", "Hormonal_contraception")) |>
    select(ID, Condition, Ovulating, Hormonal_contraception) |>
    pivot_longer(Ovulating:Hormonal_contraception,
      names_to = "Variable",
      values_to = "Value") |>
    mutate(Variable = str_replace_all(Variable, "_", " ")) |>
    ggplot(aes(y = Value, fill = Condition, color = Condition)) +
    geom_bar(alpha = 0.3, position = position_dodge()) +
    geom_text(aes(label = scales::percent(after_stat(prop)), accuracy = 0.1)),
      vjust = "inward",
      position = position_dodge(.9),
      stat = "prop",
      color = "black",
      size = 2.5) +
    facet_wrap(~Variable, scales = "free") +
```

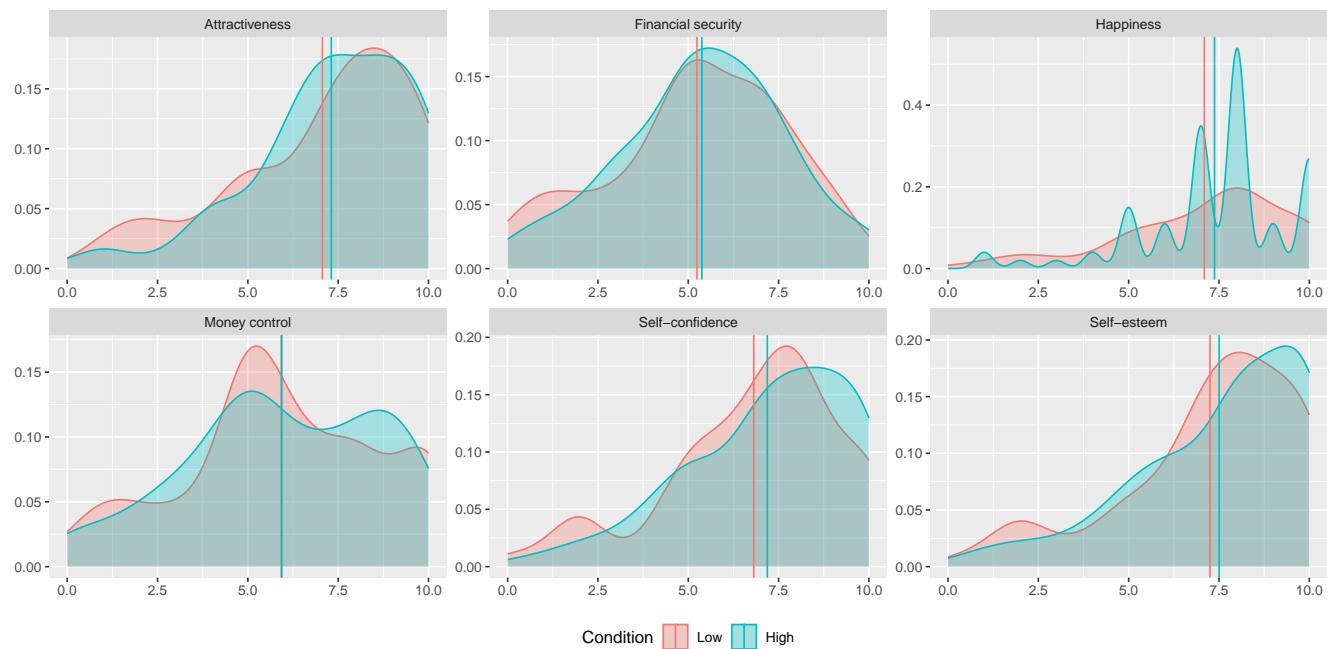
```
scale_y_discrete(labels = label_wrap(20)) +
  theme(axis.text.y = element_text(size = 8)) +
  labs(x = NULL, y = NULL),
widths = c(1, 2),
common.legend = TRUE,
legend = "bottom",
labels = "auto")
```



**Figure S8.** Distribution of values across hormonal variables, by condition. **a.** Distribution of values for body temperature. Colored vertical lines indicate the mean value for participants in each condition. **b.** Proportional number of participants across categorical variables.

### 2.3.6 Self-perceived conditions

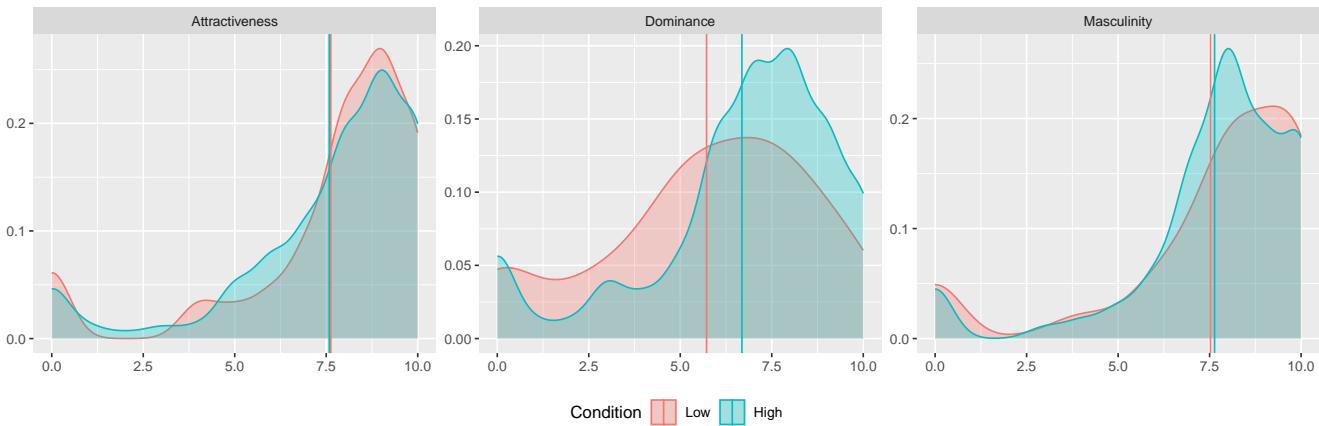
```
desc_quest |>
  select(ID, Condition, starts_with("SP_"), -SP_health) |>
  pivot_longer(where(is.numeric),
              names_to = "Variable",
              values_to = "Value") |>
  mutate(Variable = str_replace_all(Variable, "SP_", ""))
  mutate(Variable = str_replace_all(Variable, "self_", "self-"))
  mutate(Variable = str_replace_all(Variable, "_", " "))
  mutate(Variable = str_to_sentence(Variable)) |>
  ggplot(aes(x = Value, fill = Condition, color = Condition)) +
  geom_density(alpha = 0.3) +
  theme(legend.position = "bottom") +
  stat_summary(aes(xintercept = after_stat(x), y = 0),
               fun = mean, geom = "vline", orientation = "y") +
  labs(x = NULL, y = NULL) +
  facet_wrap(~Variable, scales = "free")
```



**Figure S9.** Distribution of values across self-perceived conditions. Colored vertical lines indicate the mean value for participants in each condition.

### 2.3.7 Current/last partner perception

```
desc_quest |>
  select(ID, Condition, Partner_masculinity, Partner_dominance,
         Partner_attractiveness) |>
  pivot_longer(where(is.numeric),
              names_to = "Variable",
              values_to = "Value") |>
  mutate(Variable = str_replace_all(Variable, "Partner_", "")) |>
  mutate(Variable = str_to_sentence(Variable)) |>
  ggplot(aes(x = Value, fill = Condition, color = Condition)) +
  geom_density(alpha = 0.3) +
  theme(legend.position = "bottom") +
  stat_summary(aes(xintercept = after_stat(x), y = 0),
               fun = mean, geom = "vline", orientation = "y") +
  labs(x = NULL, y = NULL) +
  facet_wrap(~Variable, scales = "free")
```



**Figure S10.** Distribution of values across perceptions of the last partner by condition. Colored vertical lines indicate the mean value for participants in each condition.

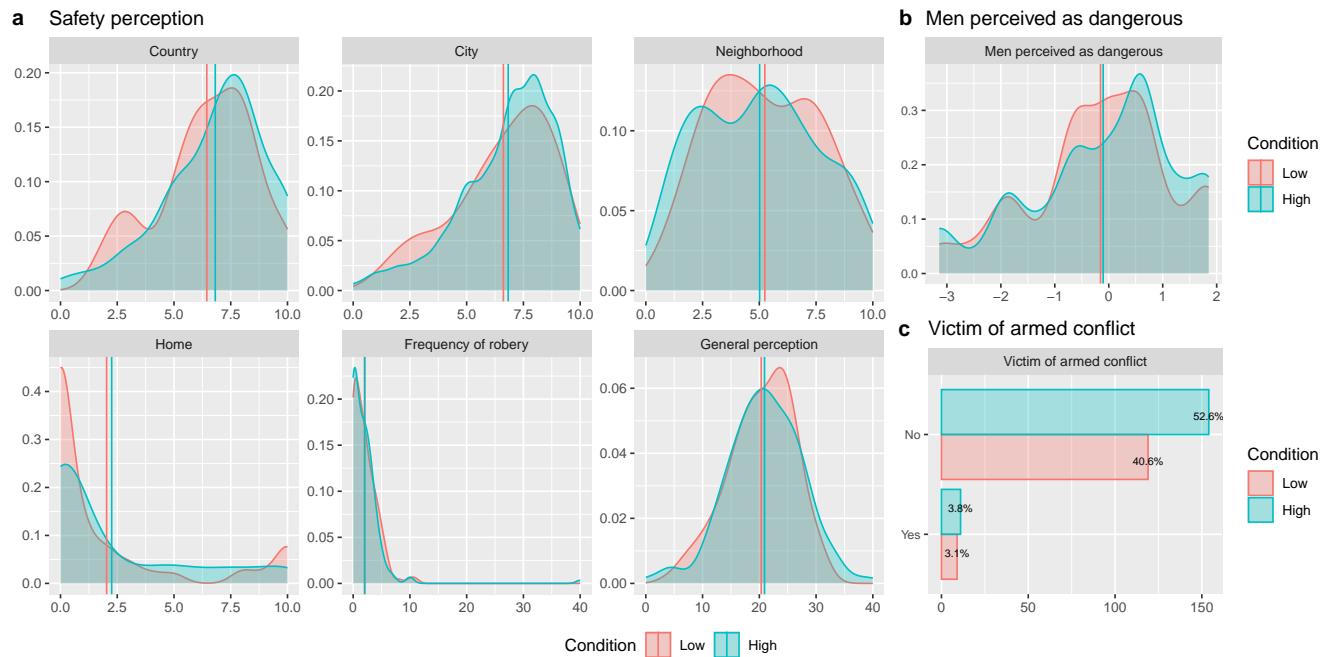
### 2.3.8 Context violence

```
ggarrange(desc_quest |>
  select(ID, Condition, ends_with("_safety"), Freq_robbery) |>
  pivot_longer(where(is.numeric),
              names_to = "Variable",
              values_to = "Value") |>
  mutate(Value = as.numeric(Value)) |>
  mutate(Variable = str_replace_all(Variable, "_safety", "")) |>
  mutate(Variable = str_replace_all(Variable, "Perceived_", "")) |>
  mutate(Variable = str_replace_all(Variable, "Freq_", "Frequency of ")) |>
  mutate(Variable = str_replace_all(Variable, "Perceived", "General perception")) |>
  mutate(Variable = str_to_sentence(Variable)) |>
  ggplot(aes(x = Value, fill = Condition, color = Condition)) +
  geom_density(alpha = 0.3) +
  labs(title = "Safety perception") +
  facet_wrap(~factor(Variable, c("Country", "City", "Neighborhood", "Home",
                                "Frequency of robbery", "General perception")),
            scales = "free") +
  stat_summary(aes(xintercept = after_stat(x), y = 0),
              fun = mean, geom = "vline", orientation = "y") +
  labs(x = NULL, y = NULL),
  ggarrange(desc_quest |>
    select(ID, Condition,
           Men_perceived_as_dangerous) |>
    pivot_longer(Men_perceived_as_dangerous,
                names_to = "Variable",
                values_to = "Value") |>
    mutate(Variable = str_replace_all(Variable,
                                     "_", " ")) |>
    mutate(Variable = str_to_sentence(Variable)) |>
    ggplot(aes(x = Value, fill = Condition, color = Condition)) +
    geom_density(alpha = 0.3) +
    labs(title = "Men perceived as dangerous") +
    facet_wrap(~Variable, scales = "free") +
    stat_summary(aes(xintercept = after_stat(x), y = 0),
                fun = mean, geom = "vline", orientation = "y") +
    theme(axis.text.y = element_text(size = 8)) +
```

```

  labs(x = NULL, y = NULL),
  desc_quest |>
  select(ID, Condition, Victim_of_armed_conflict) |>
  pivot_longer(Victim_of_armed_conflict,
               names_to = "Variable",
               values_to = "Value") |>
  mutate(Variable = str_replace_all(Variable,
                                   "_", " ")) |>
  ggplot(aes(y = Value, fill = Condition, color = Condition)) +
  geom_bar(alpha = 0.3, position = position_dodge()) +
  geom_text(aes(label = scales::percent(after_stat(prop), accuracy = 0.1)),
            vjust = "inward",
            position = position_dodge(.9),
            stat = "prop",
            color = "black",
            size = 2.5) +
  labs(title = "Victim of armed conflict") +
  facet_wrap(~Variable, scales = "free") +
  scale_y_discrete(labels = label_wrap(20)) +
  theme(axis.text.y = element_text(size = 8)) +
  labs(x = NULL, y = NULL),
  ncol = 1,
  labels = c("", "c")),
widths = c(2, 1),
common.legend = TRUE,
legend = "bottom",
labels = "auto")

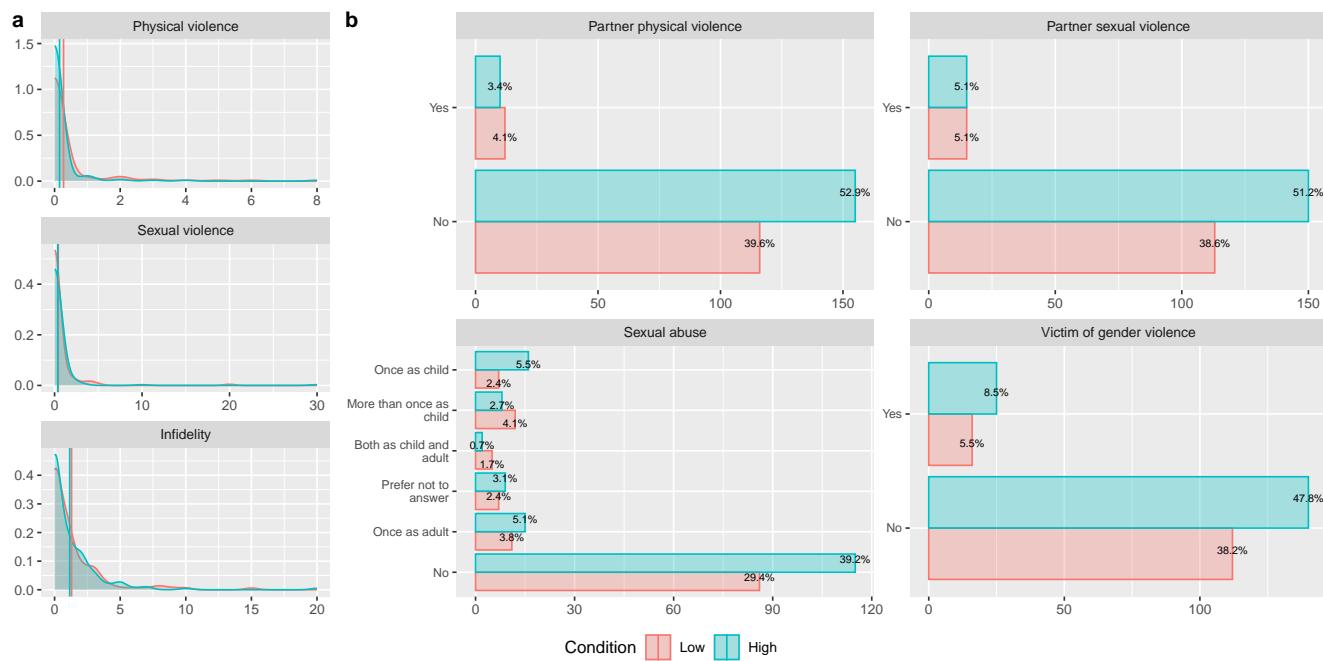
```



**Figure S11.** Distribution of values across perceptions of violence, by condition. **a.** Distribution of values across variables related to safety perception. **b.** Perceptions of men as dangerous. **c.** Proportional number of participants who reported being victims of the Colombian armed conflict. For panels a and b, colored vertical lines indicate the mean value for each variable under each condition.

### 2.3.9 Gender and partner violence

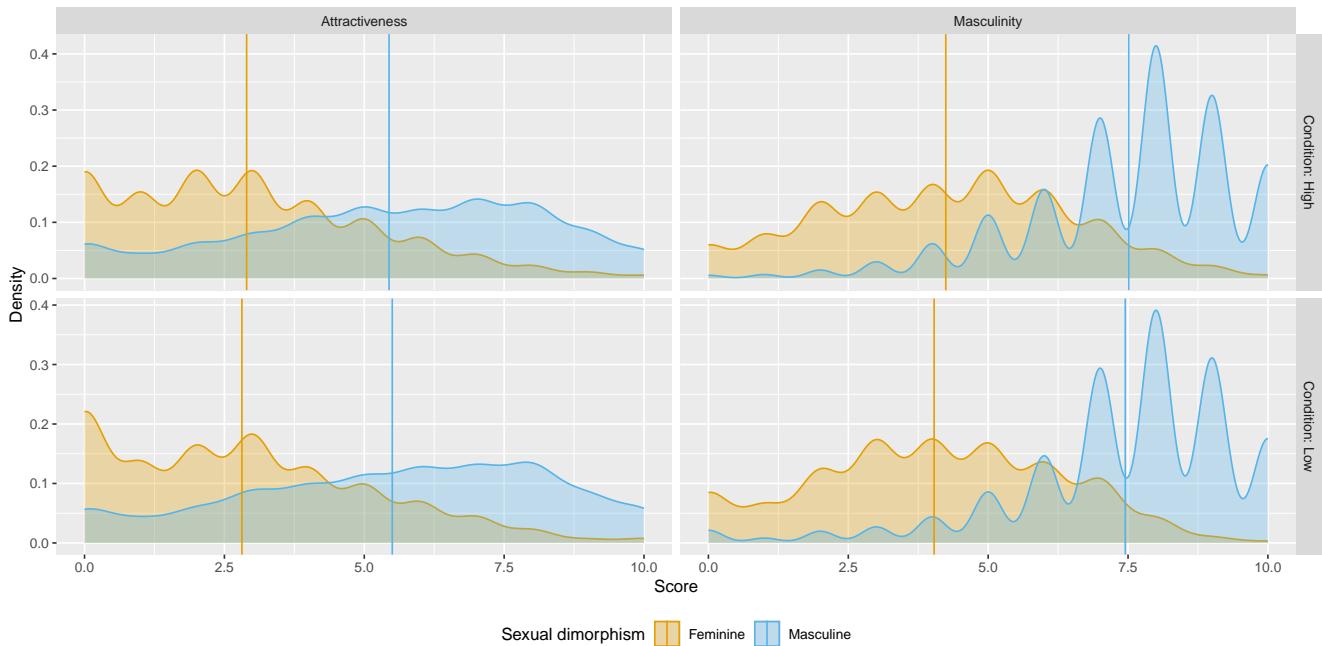
```
ggarrange(desc_quest |>
  select(ID, Condition, Freq_partner_physical_violence,
         , Freq_partner_sexual_violence, Freq_partner_infidelity) |>
  pivot_longer(where(is.numeric),
              names_to = "Variable",
              values_to = "Value") |>
  mutate(Value = as.numeric(Value)) |>
  mutate(Variable = str_replace_all(Variable, "Freq_partner_", ""))
  mutate(Variable = str_replace_all(Variable, "_", " ")) |>
  mutate(Variable = str_to_sentence(Variable)) |>
  ggplot(aes(x = Value, fill = Condition, color = Condition)) +
  geom_density(alpha = 0.3) +
  facet_wrap(~factor(Variable, c("Physical violence",
                                "Sexual violence",
                                "Infidelity")),
             scales = "free", ncol = 1) +
  stat_summary(aes(xintercept = after_stat(x), y = 0),
               fun = mean, geom = "vline", orientation = "y") +
  labs(x = NULL, y = NULL),
desc_quest |>
  select(ID, Condition,
         Victim_of_gender_violence,
         Partner_physical_violence,
         Partner_sexual_violence,
         Sexual_abuse) |>
  pivot_longer(Victim_of_gender_violence:Sexual_abuse,
              names_to = "Variable",
              values_to = "Value") |>
  mutate(Value = as.factor(Value)) |>
  mutate(Variable = str_replace_all(Variable,
                                   " ", " ")) |>
  mutate(Variable = str_to_sentence(Variable)) |>
  ggplot(aes(y = Value, fill = Condition, color = Condition)) +
  geom_bar(alpha = 0.3, position = position_dodge()) +
  geom_text(aes(label = scales::percent(after_stat(prop), accuracy = 0.1)),
            vjust = "inward",
            position = position_dodge(.9),
            stat = "prop",
            color = "black",
            size = 2.5) +
  facet_wrap(~Variable,
             scales = "free") +
  scale_y_discrete(labels = label_wrap(20)) +
  theme(axis.text.y = element_text(size = 8)) +
  labs(x = NULL, y = NULL),
widths = c(1, 3),
common.legend = TRUE,
legend = "bottom",
labels = "auto")
```



**Figure S12.** Distribution of values across gender and partner violence suffered by participants, by condition. **a.** Distribution of values across numeric variables. Colored vertical lines indicate the mean value for each variable under each condition. **b.** Proportional number of participants wacross categorical variables.

### 2.3.10 Subjective evaluation of stimuli

```
eval_long |>
  left_join(reg, by = c("ID")) |>
  filter(ID %in% unique(dat$ID)) |>
  rowwise() |>
  mutate(Sexual_dimorphism = ifelse(grepl("F", Stimulus), "Feminine", "Masculine")) |>
  select(Condition, Sexual_dimorphism, Attractiveness, Masculinity) |>
  pivot_longer(Attractiveness:Masculinity,
               names_to = "Variable",
               values_to = "Value") |>
  ggplot(aes(x = Value, fill = Sexual_dimorphism, color = Sexual_dimorphism)) +
  geom_density(alpha = 0.3) +
  theme(legend.position = "bottom") +
  labs(y = "Density", x = "Score", color = "Sexual dimorphism", fill = "Sexual dimorphism") +
  facet_grid(Condition~Variable, scales = "free",
             labeller = labeller(Condition = cond_labs)) +
  scale_color_manual(values = c("#E69F00", "#56B4E9")) +
  scale_fill_manual(values = c("#E69F00", "#56B4E9")) +
  stat_summary(aes(xintercept = after_stat(x), y = 0),
               fun = mean, geom = "vline", orientation = "y")
```



**Figure S13.** Distribution of values across subjective evaluations of attractiveness and masculinity of the stimuli used in the experiment, split by sexual dimorphism manipulations (feminine, masculine). Panels on the left are for attractiveness scores, and on the right for masculinity scores. Top panels are for participants in the high condition, and on the bottom for the low condition. Colored vertical lines indicate the mean value for participants in each condition.

## 2.4 Correlations

### 2.4.1 Correlations between partner violence and responses to masculinized and feminized stimuli

```
dat_PCF <- dat |>
  mutate(CF = as.numeric(recode(CF,
    "Yes" = "1",
    "No" = "0"))) |>
  group_by(ID, Sexual_dimorphism, Relationship, Condition) |>
  summarise(CF = sum(CF)) |>
  group_by(ID, Sexual_dimorphism, Relationship, Condition) |>
  summarise(PCF = sum(CF)/30) |>
  ungroup()

dat_short <- dat |>
  group_by(ID, Relationship, Condition, Sexual_dimorphism) |>
  summarise(Freq_partner_physical_violence = mean(Freq_partner_physical_violence),
            Freq_partner_sexual_violence = mean(Freq_partner_sexual_violence),
            Freq_partner_infidelity = mean(Freq_partner_infidelity),
            Men_perceived_as_dangerous = mean(Men_perceived_as_dangerous),
            Perceived_home_safety = mean(Perceived_home_safety),
            DFF = mean(DFF),
            TFD = mean(TFD),
            NF = mean(NF),
            Attr = mean(Attractiveness),
            Masc = mean(Masculinity)) |>
  left_join(dat_PCF |>
    select(ID, Sexual_dimorphism, Relationship, Condition, PCF),
    by = c("ID", "Sexual_dimorphism", "Relationship", "Condition")) |>
```

```

rename("PPV" = "Freq_partner_physical_violence",
      "PSV" = "Freq_partner_sexual_violence",
      "PI" = "Freq_partner_infidelity",
      "MPD" = "Men_perceived_as_dangerous",
      "PHS" = "Perceived_home_safety") |>
ungroup()

left_join(
  dat_short |>
  filter(Relationship == "Short term" &
         Condition == "Low" &
         Sexual_dimorphism == "Masculinized") |>
  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  dat_short |>
  filter(Relationship == "Short term" &
         Condition == "Low" &
         Sexual_dimorphism == "Feminized") |>
  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  by = "Effect") |>
bind_rows(left_join(
  dat_short |>
  filter(Relationship == "Long term" &
         Condition == "Low" &
         Sexual_dimorphism == "Masculinized") |>
  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  dat_short |>
  filter(Relationship == "Long term" &
         Condition == "Low" &
         Sexual_dimorphism == "Feminized") |>
  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  by = "Effect")) |>
bind_rows(left_join(
  dat_short |>
  filter(Relationship == "Short term" &
         Condition == "High" &
         Sexual_dimorphism == "Masculinized") |>
  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  dat_short |>
  filter(Relationship == "Short term" &
         Condition == "High" &
         Sexual_dimorphism == "Feminized") |>

```

```

  select(where(is.numeric)) |>
  corr.stars() |>
  rownames_to_column(var = "Effect") |>
  dplyr::slice(-1),
  by = "Effect")) |>
bind_rows(left_join(
  dat_short |>
  filter(Relationship == "Long term" &
         Condition == "High" &
         Sexual_dimorphism == "Masculinized") |>
select(where(is.numeric)) |>
corr.stars() |>
rownames_to_column(var = "Effect") |>
dplyr::slice(-1),
dat_short |>
  filter(Relationship == "Long term" &
         Condition == "High" &
         Sexual_dimorphism == "Feminized") |>
select(where(is.numeric)) |>
corr.stars() |>
rownames_to_column(var = "Effect") |>
dplyr::slice(-1),
by = "Effect")) |>
kable(digits = 2,
      booktabs = TRUE,
      align = c("l", rep("c", 20)),
      linesep = "",
      col.names = c("Effect",
                   rep(c("PPV", "PSV", "PI", "MPD", "PHS", "DFF",
                         "TDF", "NF", "Attr.", "Masc."), times = 2)),
      caption = "Correlations between partner violence and responses to masculinized and
feminized stimuli",
      escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
add_header_above(c(" ", "Masculinized" = 10,
                  "Feminized" = 10),
                 bold = TRUE) |>
pack_rows("Low condition, short-term relationship", 1, 10,
          hline_after = TRUE) |>
pack_rows("Low condition, long-term relationship", 11, 20,
          hline_after = TRUE, hline_before = TRUE) |>
pack_rows("High condition, short-term relationship", 21, 30,
          hline_after = TRUE, hline_before = TRUE) |>
pack_rows("High condition, long-term relationship", 31, 40,
          hline_after = TRUE, hline_before = TRUE) |>
footnote(general = paste0("Values represent Pearson correlation coefficients ($r$). ",
                           "PPV = Frequency of partner physical violence;",
                           "PSV = Frequency of partner sexual violence;",
                           "PI = Frequency of partner infidelity;",
                           "MPD = Men perceived as dangerous;",
                           "PHS = Perceived home safety;",
                           "DFF = Duration of First Fixation;",
                           "TDF = Total Fixation Duration;",
                           "NF = Number of Fixations;",
                           "Attr. = attractiveness;"))

```

```

Masc. = masculinity;
PCF = proportion of Chosen Faces. ",
"For significance, $^{\dagger} < 0.1, *$p$ < 0.05, ",
**$p$ < 0.01, ***$p$ < 0.001. ",
"Significant correlations are in bold."),
threeparttable = TRUE,
footnote_as_chunk = TRUE,
escape = FALSE)

```

**Table S5.** Correlations between partner violence and responses to masculinized and feminized stimuli

Effect	Masculinized									Feminized										
	PPV	PSV	PI	MPD	PHS	DFF	TFD	NF	Attr.	Masc.	PPV	PSV	PI	MPD	PHS	DFF	TFD	NF	Attr.	Masc.
<b>Low condition, short-term relationship</b>																				
PSV	<b>0.31***</b>										<b>0.31***</b>									
PI	<b>0.38***</b>	0.07									<b>0.38***</b>	0.07								
MPD	0.02	<b>0.22*</b>	-0.08								0.02	<b>0.22*</b>	-0.08							
PHS	-0.06	-0.10	-0.07	0.11							-0.06	-0.10	-0.07	0.11						
DFF	<b>-0.21*</b>	-0.04	0.02	<b>0.22*</b>	-0.01						<b>-0.23*</b>	-0.01	-0.09	<b>0.21*</b>	0.02					
TFD	0.11	0.06	0.06	0.02	0.05	-0.04					0.05	0.14	-0.08	0.17 <sup>†</sup>	-0.01	-0.09				
NF	<b>0.26**</b>	0.07	0.09	-0.02	0.06	<b>-0.56***</b>	<b>0.83***</b>				0.17 <sup>†</sup>	0.12	0.00	0.09	-0.01	<b>-0.54***</b>	<b>0.88***</b>			
Attr	-0.06	-0.13	0.07	-0.09	0.07	-0.08	-0.09	-0.04			-0.09	-0.13	0.01	0.07	-0.08	-0.06	0.14	0.13		
Masc	-0.02	-0.01	0.09	-0.09	0.06	-0.07	-0.11	-0.08	<b>0.62***</b>		0.03	-0.04	0.00	-0.12	-0.02	-0.02	0.09	0.05	<b>0.63***</b>	
PCF	0.15 <sup>†</sup>	0.01	0.06	-0.02	0.04	0.10	<b>0.51***</b>	<b>0.3***</b>	0.07	-0.08	-0.15 <sup>†</sup>	-0.01	-0.16 <sup>†</sup>	0.04	-0.03	<b>-0.21*</b>	<b>0.41***</b>	<b>0.37***</b>	<b>0.22*</b>	<b>0.23*</b>
<b>Low condition, long-term relationship</b>																				
PSV	<b>0.31***</b>										<b>0.31***</b>									
PI	<b>0.38***</b>	0.07									<b>0.38***</b>	0.07								
MPD	0.02	<b>0.22*</b>	-0.08								0.02	<b>0.22*</b>	-0.08							
PHS	-0.06	-0.10	-0.07	0.11							-0.06	-0.10	-0.07	0.11						
DFF	-0.14	0.01	-0.05	<b>0.23*</b>	0.15						-0.15	0.00	-0.04	0.07	-0.04					
TFD	0.15 <sup>†</sup>	0.05	0.03	0.06	0.14	-0.09					0.05	<b>0.19*</b>	0.02	<b>0.19*</b>	-0.06	<b>0.21*</b>				
NF	<b>0.26**</b>	0.09	0.08	0.01	0.10	<b>-0.54***</b>	<b>0.84***</b>				0.15 <sup>†</sup>	0.13	0.06	0.17 <sup>†</sup>	-0.05	<b>-0.29**</b>	<b>0.88***</b>			
Attr	-0.06	-0.13	0.07	-0.09	0.07	-0.13	-0.07	-0.04			-0.09	-0.13	0.01	0.07	-0.08	-0.14	0.02	0.06		
Masc	-0.02	-0.01	0.09	-0.09	0.06	-0.11	-0.12	-0.05	<b>0.62***</b>		0.03	-0.04	0.00	-0.12	-0.02	-0.07	-0.06	-0.09	<b>0.63***</b>	
PCF	0.01	-0.03	-0.15 <sup>†</sup>	-0.02	<b>0.19*</b>	0.05	<b>0.31***</b>	0.17 <sup>†</sup>	0.14	-0.02	0.01	0.03	0.06	0.04	-0.16 <sup>†</sup>	0.05	<b>0.48***</b>	<b>0.4***</b>	<b>0.19*</b>	0.09
<b>High condition, short-term relationship</b>																				
PSV	0.09										0.09									
PI	0.13	-0.03									0.13	-0.03								
MPD	0.12	0.09	<b>0.15*</b>								0.12	0.09	<b>0.15*</b>							
PHS	0.08	-0.07	0.02	0.06							0.08	-0.07	0.02	0.06						
DFF	-0.09	-0.05	0.03	-0.01	0.06						-0.08	-0.02	0.05	0.00	0.03					
TFD	0.00	0.06	0.03	-0.05	-0.11	-0.03					<b>0.15*</b>	<b>0.14<sup>†</sup></b>	-0.02	0.07	0.04	0.14				
NF	0.04	0.12	0.03	0.00	-0.08	<b>-0.59***</b>	<b>0.81***</b>				0.13	<b>0.16*</b>	-0.02	0.08	0.03	<b>-0.43***</b>	<b>0.85***</b>			
Attr	0.00	0.03	0.09	0.03	0.07	-0.17 <sup>†</sup>	0.14 <sup>†</sup>	<b>0.17*</b>			0.00	-0.01	-0.04	0.05	0.15 <sup>†</sup>	-0.07	0.06	0.07		
Masc	0.03	0.03	0.00	-0.06	-0.01	0.02	0.05	0.15 <sup>†</sup>	0.13	<b>0.45***</b>	0.10	-0.08	-0.07	-0.05	0.08	0.14	0.09	0.00	<b>0.58***</b>	
PCF	-0.08	0.00	0.04	-0.06	-0.03	-0.03	<b>0.49***</b>	<b>0.38***</b>	<b>0.24**</b>	<b>0.19*</b>	0.11	0.02	-0.01	0.07	0.05	<b>0.23*</b>	<b>0.43***</b>	<b>0.26***</b>	0.04	0.03
<b>High condition, long-term relationship</b>																				
PSV	0.09										0.09									
PI	0.13	-0.03									0.13	-0.03								
MPD	0.12	0.09	<b>0.15*</b>								0.12	0.09	<b>0.15*</b>							
PHS	0.08	-0.07	0.02	0.06							0.08	-0.07	0.02	0.06						
DFF	-0.10	-0.09	0.11	0.00	0.06						-0.16 <sup>†</sup>	-0.10	0.05	-0.03	0.14					
TFD	-0.07	0.04	-0.05	-0.05	<b>-0.17*</b>	0.02					<b>0.2*</b>	0.12	-0.03	0.10	-0.07	-0.04				
NF	-0.02	0.11	-0.05	0.00	-0.15 <sup>†</sup>	<b>-0.52***</b>	<b>0.82***</b>				0.13	<b>0.15*</b>	-0.04	0.10	-0.08	<b>-0.52***</b>	<b>0.87***</b>			
Attr	0.00	0.03	0.09	0.03	0.07	-0.08	0.09	0.15 <sup>†</sup>			0.00	-0.01	-0.04	0.05	0.15 <sup>†</sup>	-0.04	0.09	0.07		
Masc	0.03	0.03	0.00	-0.06	-0.01	-0.04	0.06	0.09	<b>0.45***</b>		0.10	-0.08	-0.07	-0.05	0.08	0.16 <sup>†</sup>	<b>0.21**</b>	0.09	<b>0.58***</b>	
PCF	<b>-0.21**</b>	0.00	0.04	-0.13	-0.06	0.10	<b>0.32***</b>	<b>0.18*</b>	<b>0.32***</b>	0.03	<b>0.23**</b>	0.02	-0.01	<b>0.16*</b>	0.07	0.10	<b>0.44***</b>	<b>0.31***</b>	<b>0.22**</b>	0.2*

Note: Values represent Pearson correlation coefficients ( $r$ ). PPV = Frequency of partner physical violence; PSV = Frequency of partner sexual violence; PI = Frequency of partner infidelity; MPD = Men perceived as dangerous; PHS = Perceived home safety; DFF = Duration of First Fixation; TDF = Total Fixation Duration; NF = Number of Fixations; Attr. = attractiveness; Masc. = masculinity; PCF = proportion of Chosen Faces. For significance,  $^{\dagger}p < 0.1$ ,  $*p < 0.05$ ,  $**p < 0.01$ ,  $***p < 0.001$ . Significant correlations are in bold.

### 3 Manipulation check

#### 3.1 Resource availability dimensions by condition

```

reg_fin |>
  select(starts_with("Condition")) |>
  pivot_longer(cols = contains("_"),
               names_to = "Dimension",
               values_to = "Score") |>
  group_by(Dimension) |>
  summarise("Mean (Low)" = mean(Score[reg_fin$Condition == "Low"]),
            "Mean (High)" = mean(Score[reg_fin$Condition == "High"]),
            "$t$" = t.test(Score ~ Condition)$statistic,
            "$p$" = t.test(Score ~ Condition)$p.value,
            "$g$" = hedges_g(Score ~ Condition)$Hedges_g) |>
  ungroup() |>

```

```

    mutate(Dimension = str_replace_all(Dimension, "_", " "),
           Dimension = str_remove_all(Dimension, "Condition "),
           Dimension = str_to_title(Dimension)) |>
  mutate("$p$" = pval.lev("$p$")) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 5)),
        linesep = "",
        caption = "Mean scores and comparison of resource availability dimensions by condition",
        escape = FALSE) |>
  kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
  add_header_above(c("",
                     "Condition" = 2,
                     " " = 3),
                   bold = TRUE) |>
  footnote(general = "Results are from Welch's \\\textit{t}-test.
As effect size, Hedges' \\\textit{g} values are reported.",
            threeparttable = TRUE,
            footnote_as_chunk = TRUE,
            escape = FALSE)

```

**Table S6.** Mean scores and comparison of resource availability dimensions by condition

Dimension	Condition		<i>t</i>	<i>p</i>	<i>g</i>
	Mean (Low)	Mean (High)			
Economic Security	0.76	6.62	54.04	< 0.0001	6.56
Happiness	1.44	6.37	39.49	< 0.0001	4.84
Healthy	2.04	6.42	30.39	< 0.0001	3.81
Physical Safety	1.90	6.27	27.36	< 0.0001	3.42

*Note:* Results are from Welch's *t*-test. As effect size, Hedges' *g* values are reported.

### 3.2 Effect of sexual dimorphism manipulation on masculinity and attractiveness ratings, by condition

```

# Clean data fro models
eval_desc <- dat |>
  group_by(ID, Sexual_dimorphism, Condition) |>
  summarise(Masculinity = mean(Masculinity),
            Attractiveness = mean(Attractiveness))

# Masculinity
mod_masc <- lmer(Masculinity ~ Sexual_dimorphism * Condition + (1 | ID), data = eval_desc)
# anova(mod_masc)
contr_mod_masc <- as.data.frame(pairs(emmeans(mod_masc,
                                                ~ Sexual_dimorphism | Condition))) |>
  separate(contrast, c("group1", "group2"), " - ") |>
  mutate(p.signif = stars.pval(p.value))

p_mancheck_masc <- ggplot(eval_desc, aes(x = Sexual_dimorphism,
                                             y = Masculinity,
                                             color = Sexual_dimorphism)) +
  geom_jitter(alpha = 0.5) +
  stat_summary(fun.data = "mean_cl_boot",

```

```

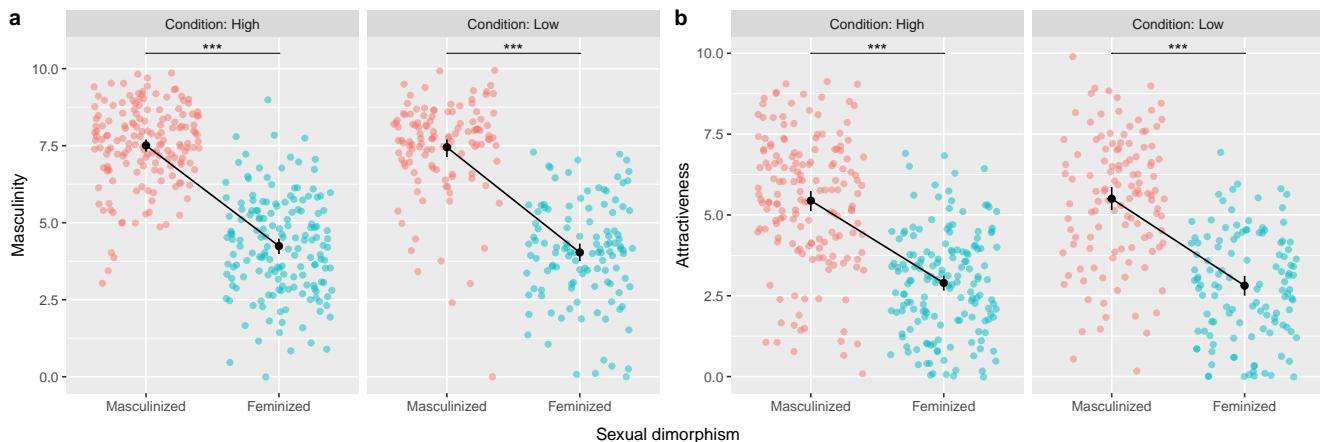
        color = "black",
        size = 0.3) +
stat_summary(fun.y = mean,
            colour = "black",
            geom = "line",
            aes(group = 1)) +
stat_pvalue_manual(contr_mod_masc, label = "p.signif",
                    y.position = 10.5,
                    hide.ns = TRUE,
                    tip.length = 0) +
labs(x = NULL,
      color = "Sexual dimorphism") +
facet_wrap(~Condition,
           labeller = labeller(Condition = cond_labs))

# Attractiveness
mod_attr <- lmer(Attractiveness ~ Sexual_dimorphism * Condition + (1 | ID), data = eval_desc)
# anova(mod_attr)
contr_mod_attr <- as.data.frame(pairs(emmeans(mod_attr,
                                                ~ Sexual_dimorphism | Condition))) |>
separate(contrast, c("group1", "group2"), " - ") |>
mutate(p.signif = stars.pval(p.value))

p_mancheck_attr <- ggplot(eval_desc, aes(x = Sexual_dimorphism,
                                            y = Attractiveness,
                                            color = Sexual_dimorphism)) +
geom_jitter(alpha = 0.5) +
stat_summary(fun.data = "mean_cl_boot",
            color = "black",
            size = 0.3) +
stat_summary(fun.y = mean, colour = "black", geom = "line", aes(group = 1)) +
stat_pvalue_manual(contr_mod_attr, label = "p.signif",
                    y.position = 10,
                    hide.ns = TRUE,
                    tip.length = 0) +
labs(x = NULL,
      color = "Sexual dimorphism") +
facet_wrap(~Condition,
           labeller = labeller(Condition = cond_labs))

# Combined plot
ggarrange(p_mancheck_masc, p_mancheck_attr,
           legend = "none",
           labels = "auto") |>
annotate_figure(bottom = text_grob("Sexual dimorphism",
                                    size = 11))

```



**Figure S14.** Effect of sexual dimorphism manipulation on ratings of (a) masculinity and (b) attractiveness, by condition (High, Low). Reported significance are contrasts between sexual dimorphism levels (feminized, masculinized) from linear mixed models including the fixed effects of condition, sexual dimorphism, and their interaction, as well as random intercepts per participants. In both models the main (within-subject) effect of the sexual dimorphism manipulation was significant, but not the main effect of condition or its interaction with sexual dimorphism. Mean ratings and 95% CIs are in black. \*\*\* $p < 0.001$ .

## 4 Models of the experimental design

We modeled the experimental design, by fitting linear models with the main effects and all possible interactions between Sexual dimorphism (masculinized, feminized), Relationship (short term, long term), and Condition (low, high). Separate models were fitted for each dependent variable: Duration of First Fixations (DFF), Total Fixation Duration (TFD), Number of Fixations (NF), and Proportion of Chosen Faces (PCF).

For models of fixations (DFF, TFD, and NF) we included random intercepts and random slopes for each participant between sexual dimorphism levels, as well as random intercepts for each stimulus (i.e. these were linear mixed models), in line with recommendations (Barr et al., 2013). However, to test the Proportion of Chosen Faces (PCF), we used a linear model without random effects, as there was only one value per participant (a proportion) for each combination of factors.

As effect sizes, for linear mixed models we report Nakagawa's conditional and marginal  $R^2$  values (Nakagawa & Schielzeth, 2013), but we do not report standardized effect sizes for individual terms, as there is no accepted method to do so.

For the linear model of Proportion of Chosen Faces (PCF), we report both unadjusted and adjusted  $R^2$  as full model effect sizes, but also include  $\omega_p^2$  scores as standardized effect sizes for each term. As frequently reported,  $\omega^2$  is a more reliable measure of effect size than the more commonly used  $\eta^2$  (e.g., Kroes & Finley, 2023).

### 4.1 Duration of First Fixations (DFF)

#### 4.1.1 Data

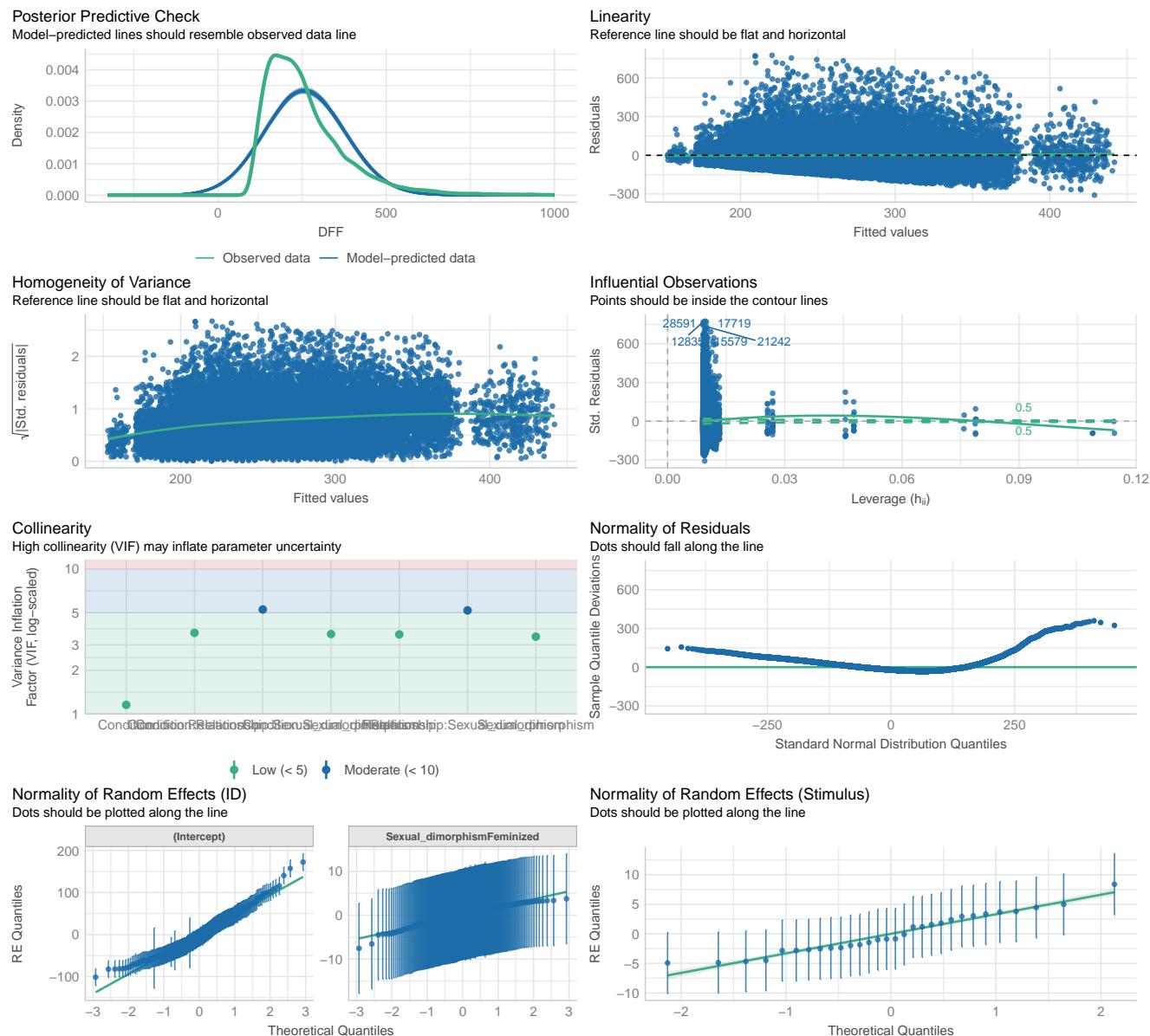
```
dat_m1 <- dat |>
  select(DFF, Condition, Relationship, Sexual_dimorphism,
         ID, Stimulus,
         Freq_partner_physical_violence, Freq_partner_sexual_violence,
         Freq_partner_infidelity, Men_perceived_as_dangerous,
         Perceived_home_safety) |>
  filter(DFF >= 100 & DFF <= 1000) |>
  drop_na()
```

#### 4.1.2 Fit linear mixed model

```
mod1 <- lmer(DFF ~ Condition * Relationship * Sexual_dimorphism +
               (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
               data = dat_m1)
```

**4.1.2.1 Model assumptions** Model assumptions were checked using the `check_model` function from the `performance` package (Lüdecke et al., 2021).

```
check_model(mod1)
```



**Figure S15.** Model assumptions. Plots represent prediction check, linearity, homogeneity of variance, influential observations, collinearity, and normality of both residuals and random effects, respectively.

#### 4.1.3 Table of fixed effects

```
tab_m1 <- lmer.anova.tab(model = mod1)
tab_m1$kab
```

**Table S7.** ANOVA-type table of fixed effects for the DFF model

Fixed effect	F	df	p
Condition	0.06	1, 290	0.8
Relationship	0.08	1, 33368.84	0.77
Sexual dimorphism	10.91	1, 285.13	<b>0.0011</b>
Condition × Relationship	0.14	1, 33369.04	0.71
Condition × Sexual dimorphism	0.01	1, 285.13	0.93
Relationship × Sexual dimorphism	0.46	1, 33424.64	0.5
Condition × Relationship × Sexual dimorphism	1.25	1, 33424.51	0.26
<b>Nakagawa's R<sup>2</sup></b>			
Conditional	0.17		
Marginal	0.00038		

#### 4.1.4 Estimated marginal means and *post-hoc* contrasts of significant effects

**4.1.4.1 Main effect: Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli.

```
main.eff.contr(model = mod1,
  emm_contr = emmeans(mod1, pairwise ~ Sexual_dimorphism))
```

**Table S8.** Estimated marginal and contrast between Sexual dimorphism levels for the DFF model

Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
					Difference	SE	z	p
Masculinized	260.65	3.10	254.57	266.73	4.08	1.23	3.3	< 0.001
Feminized	256.57	3.04	250.62	262.52				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.1.5 Full design contrasts

Table of estimated marginal means and contrasts between masculinized and feminized stimuli including the full experimental design.

```
full.contr(model = mod1,
  emm_contr = emmeans(mod1, pairwise ~ Sexual_dimorphism | Relationship + Condition))
```

**Table S9.** Estimated marginal and contrast between masculinized and feminized stimuli by Relationship and Condition for the DFF model

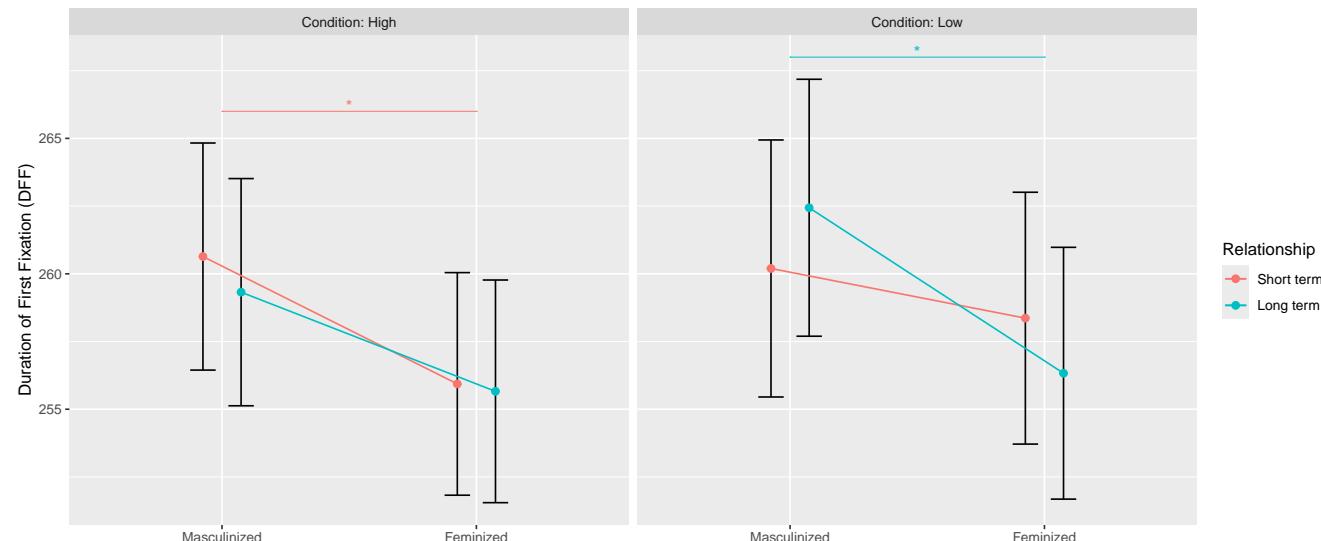
Condition	Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
							Difference	SE	z	p
High	Short term	Masculinized	260.64	4.19	252.42	268.85	4.70	2.26	2.08	<b>0.0375</b>
		Feminized	255.93	4.11	247.88	263.99				
	Long term	Masculinized	259.32	4.19	251.10	267.54	3.66	2.27	1.61	0.11
		Feminized	255.66	4.11	247.60	263.72				
Low	Short term	Masculinized	260.20	4.75	250.90	269.50	1.83	2.57	0.71	0.48
		Feminized	258.36	4.65	249.25	267.48				
	Long term	Masculinized	262.44	4.74	253.14	271.74	6.11	2.57	2.37	<b>0.0176</b>
		Feminized	256.33	4.65	247.22	265.44				

Note: EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.1.6 Figure for the DFF model

This figure summarizes the the DFF model results.

```
pm1 <- plot.exp(model = mod1,
                  y.pos = c(266, NA, NA, 268),
                  y.lab = "Duration of First Fixation (DFF)")
pm1
```



**Figure S16.** Differences in the Duration of First Fixations (DFF) to masculinized and feminized stimuli, by relationship (short term, long term) and condition (low, high). Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S7 and S9). In all cases, significant effects are represented with lines and stars:  $*p < 0.05$ .

## 4.2 Total Fixation Duration (TFD)

### 4.2.1 Data

```
dat_m2 <- dat |>
  select(TFD, Condition, Relationship, Sexual_dimorphism,
         ID, Stimulus,
         Freq_partner_physical_violence, Freq_partner_sexual_violence,
         Freq_partner_infidelity, Men_perceived_as_dangerous,
```

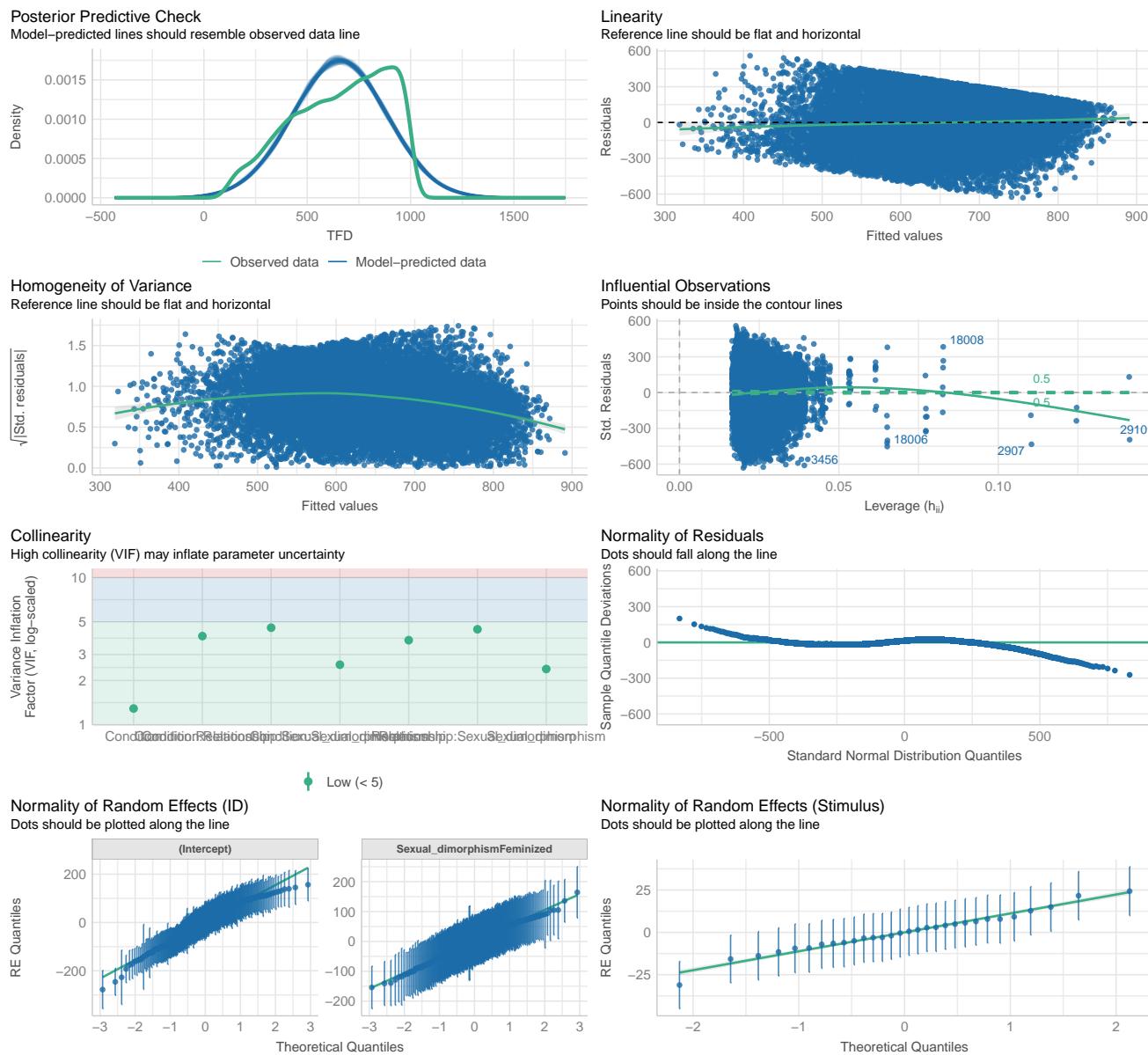
```
Perceived_home_safety) |>
filter(TFD >= 100 & TFD <= 1000) |>
drop_na()
```

#### 4.2.2 Fit linear mixed model

```
mod2 <- lmer(TFD ~ Condition * Relationship * Sexual_dimorphism +
(1 + Sexual_dimorphism | ID) + (1 | Stimulus),
data = dat_m2,
control = lmerControl(optimizer="bobyqa"))
```

**4.2.2.1 Model assumptions** Model assumptions were checked using the `check_model` function from the `performance` package (Lüdecke et al., 2021).

```
check_model(mod2)
```



**Figure S17.** Model assumptions. Plots represent prediction check, linearity, homogeneity of variance, influential observations, collinearity, and normality of both residuals and random effects, respectively.

#### 4.2.3 Table of fixed effects

```
tab_m2 <- lmer.anova.tab(model = mod2)
tab_m2$kab
```

**Table S10.** ANOVA-type table of fixed effects for the TFD model

Fixed effect	F	df	p
Condition	0.02	1, 279.87	0.88
Relationship	1.31	1, 18832.3	0.25
Sexual dimorphism	47.00	1, 279.46	< <b>0.0001</b>
Condition × Relationship	3.71	1, 18834.01	0.05
Condition × Sexual dimorphism	0.95	1, 275.61	0.33
Relationship × Sexual dimorphism	5.47	1, 18762.57	<b>0.0194</b>
Condition × Relationship × Sexual dimorphism	0.05	1, 18764.97	0.82
<b>Nakagawa's <math>R^2</math></b>			
Conditional = 0.17			
Marginal = 0.0062			

#### 4.2.4 Estimated marginal means and *post-hoc* contrasts of significant effects

**4.2.4.1 Main effect: Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli.

```
main.eff.contr(model = mod2,
                 emm_contr = emmeans(mod2, pairwise ~ Sexual_dimorphism))
```

**Table S11.** Estimated marginal and contrast between Sexual dimorphism levels for the TFD model

Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
					Difference	SE	z	p
Masculinized	685.92	6.04	674.08	697.75	34.93	5.1	6.86	< <b>0.0001</b>
Feminized	650.98	6.66	637.93	664.04				

Note: EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

**4.2.4.2 Interaction: Relationship × Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli by relationship.

```
inter.contr(model = mod2,
            emm_contr = emmeans(mod2, pairwise ~ Sexual_dimorphism | Relationship))
```

**Table S12.** Estimated marginal and contrast between Sexual dimorphism levels by Relationship for the TFD model

Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
						Difference	SE	z	p
Short term	Masculinized	684.08	6.43	671.48	696.68	27.75	5.94	4.67	< <b>0.0001</b>
	Feminized	656.33	6.99	642.63	670.03				
Long term	Masculinized	687.76	6.46	675.10	700.41	42.12	5.96	7.07	< <b>0.0001</b>
	Feminized	645.64	6.97	631.97	659.31				

Note: EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.2.5 Full design contrasts

Table of estimated marginal means and contrasts between masculinized and feminized stimuli including the full experimental design.

```
full.contr(model = mod2,
            emm_contr = emmeans(mod2, pairwise ~ Sexual_dimorphism | Relationship + Condition))
```

**Table S13.** Estimated marginal and contrast between masculinized and feminized stimuli by Relationship and Condition for the TFD model

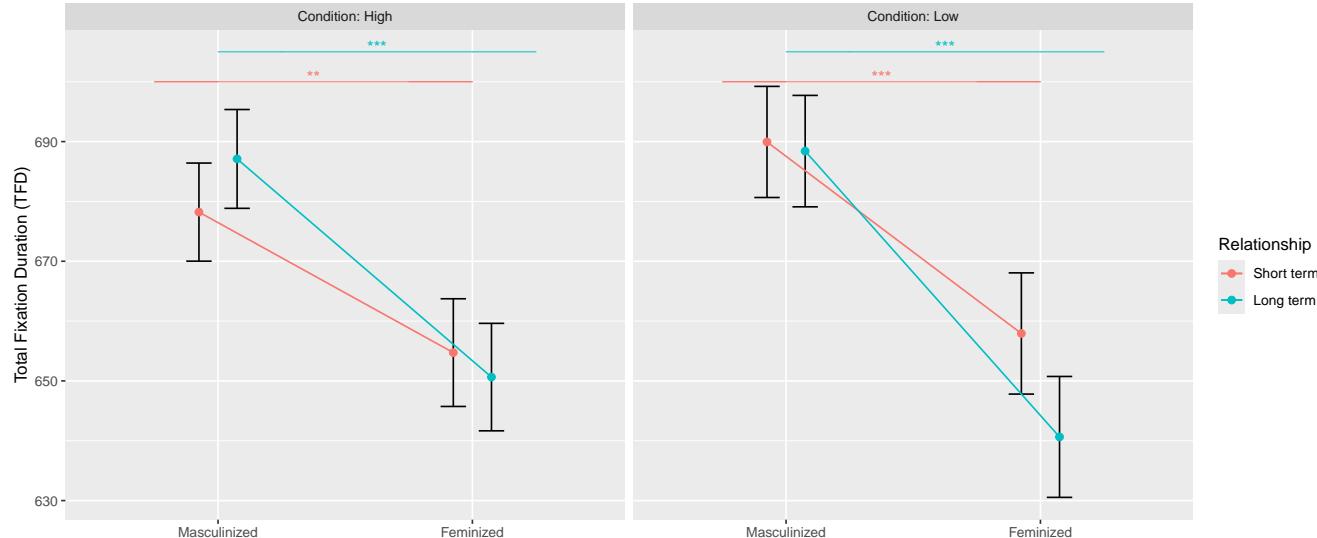
Condition	Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
							Difference	SE	z	p
High	Short term	Masculinized	678.22	8.20	662.15	694.28	23.49	7.81	3.01	<b>0.0027</b>
		Feminized	654.73	9.00	637.09	672.37				
	Long term	Masculinized	687.10	8.26	670.92	703.29	36.46	7.85	4.64	< 0.0001
		Feminized	650.64	8.98	633.04	668.24				
Low	Short term	Masculinized	689.94	9.29	671.74	708.14	32.01	8.93	3.58	< 0.001
		Feminized	657.93	10.13	638.07	677.78				
	Long term	Masculinized	688.41	9.31	670.16	706.66	47.77	8.94	5.34	< 0.0001
		Feminized	640.64	10.10	620.83	660.44				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.2.6 Figure for the TFD model

This figure summarizes the the TFD model results.

```
pm2 <- plot.exp(model = mod2,
                  y.pos = c(700, 700, 705, 705),
                  y.lab = "Total Fixation Duration (TFD)")
pm2
```



**Figure S18.** Differences in the Total Fixation Duration (TFD) to masculinized and feminized stimuli, by relationship (short term, long term) and condition (low, high). Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S10 and S13). In all cases, significant effects are represented with lines and stars: \*\* $p < 0.01$ , \*\*\* $p < 0.001$

### 4.3 Number of Fixations (NF)

#### 4.3.1 Data

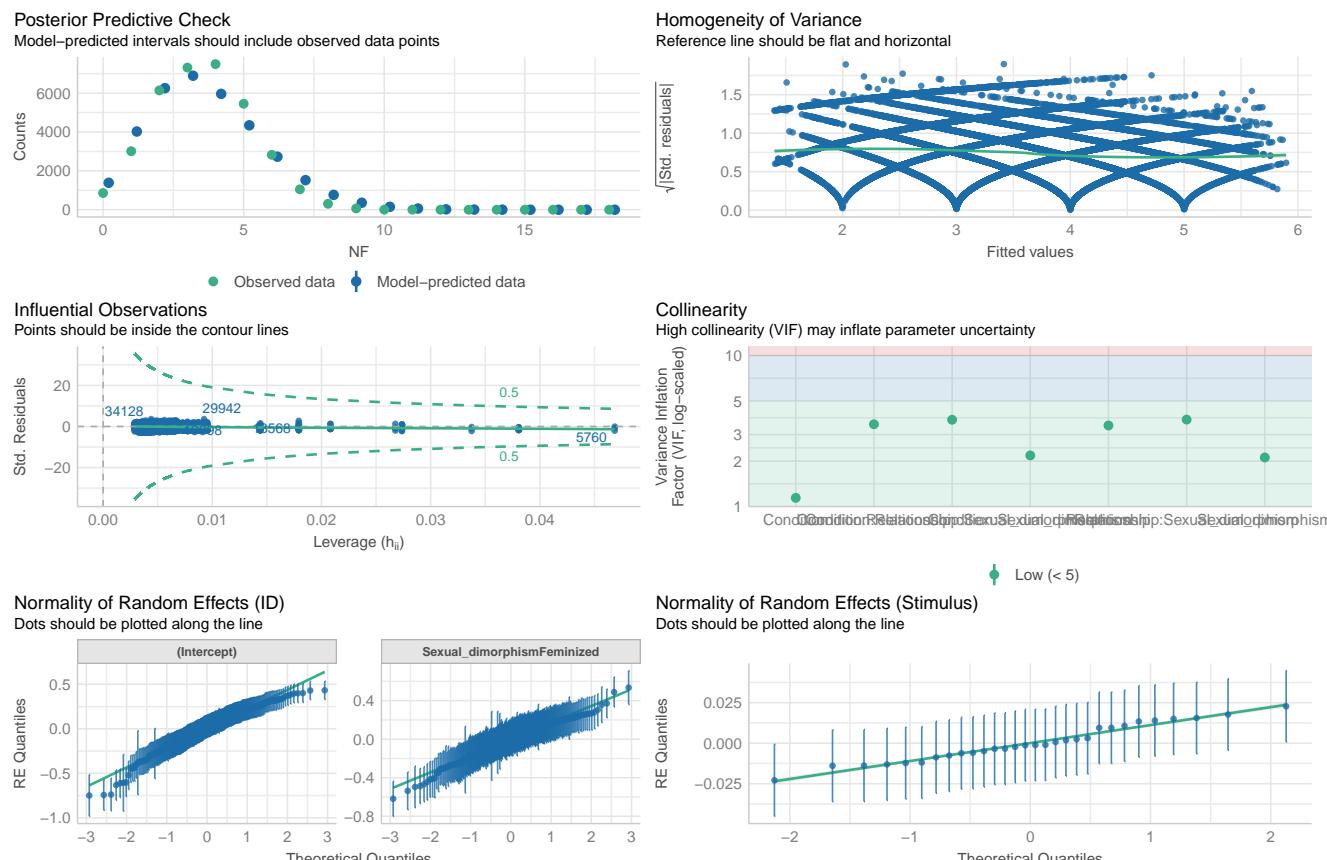
```
dat_m3 <- dat |>
  select(NF, Condition, Relationship, Sexual_dimorphism,
         ID, Stimulus,
         Freq_partner_physical_violence, Freq_partner_sexual_violence,
         Freq_partner_infidelity, Men_perceived_as_dangerous,
         Perceived_home_safety) |>
  group_by(ID, Stimulus, Relationship) |>
  filter(!sum(NF[Sexual_dimorphism == "Masculinized"] == 0) > 0 &
         sum(NF[Sexual_dimorphism == "Feminized"] == 0) > 0)) |>
  ungroup() |>
  drop_na()
```

#### 4.3.2 Fit linear mixed model

```
mod3 <- glmer(NF ~ Condition * Relationship * Sexual_dimorphism +
               (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
               data = dat_m3,
               family = poisson)
```

**4.3.2.1 Model assumptions** Model assumptions were checked using the `check_model` function from the `performance` package (Lüdecke et al., 2021).

```
check_model(mod3, check = c("pp_check", "homogeneity", "outliers", "vif", "reqq"))
```



**Figure S19.** Model assumptions. Plots represent prediction check, homogeneity of variance, influential observations, collinearity, and normality of random effects, respectively.

### 4.3.3 Table of fixed effects

```
tab_m3 <- glmer.anova.tab(model = mod3)
tab_m3$kab
```

**Table S14.** ANOVA-type table of fixed effects for the NF model

Fixed effect	$\chi^2$	df	p
Condition	0.00	1	1
Relationship	0.03	1	0.87
Sexual dimorphism	38.12	1	< 0.0001
Condition × Relationship	2.56	1	0.11
Condition × Sexual dimorphism	0.78	1	0.38
Relationship × Sexual dimorphism	10.75	1	0.001
Condition × Relationship × Sexual dimorphism	0.44	1	0.51
<b>Nakagawa's <math>R^2</math></b>			
Conditional = 0.21			
Marginal = 0.0056			

### 4.3.4 Estimated marginal means and *post-hoc* contrasts of significant effects

**4.3.4.1 Main effect: Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli.

```
main.eff.contr(model = mod3,
  emm_contr = emmeans(mod3, pairwise ~ Sexual_dimorphism))
```

**Table S15.** Estimated marginal and contrast between Sexual dimorphism levels for the NF model

Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
					Difference	SE	z	p
Masculinized	1.26	0.01	1.24	1.29	0.08	0.01	6.24	< 0.0001
Feminized	1.18	0.02	1.15	1.22				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

**4.3.4.2 Interaction: Relationship × Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli by relationship.

```
inter.contr(model = mod3,
  emm_contr = emmeans(mod3, pairwise ~ Sexual_dimorphism | Relationship))
```

**Table S16.** Estimated marginal and contrast between Sexual dimorphism levels by Relationship for the NF model

Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
						Difference	SE	z	p
Short term	Masculinized	1.26	0.02	1.23	1.29	0.06	0.01	4.43	< 0.0001
	Feminized	1.19	0.02	1.16	1.23				
Long term	Masculinized	1.27	0.02	1.24	1.30	0.10	0.01	6.98	< 0.0001
	Feminized	1.17	0.02	1.14	1.21				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.3.5 Full design contrasts

Table of estimated marginal means and contrasts between masculinized and feminized stimuli including the full experimental design.

```
full.contr(model = mod3,
            emm_contr = emmeans(mod3, pairwise ~ Sexual_dimorphism | Relationship + Condition))
```

**Table S17.** Estimated marginal and contrast between masculinized and feminized stimuli by Relationship and Condition for the NF model

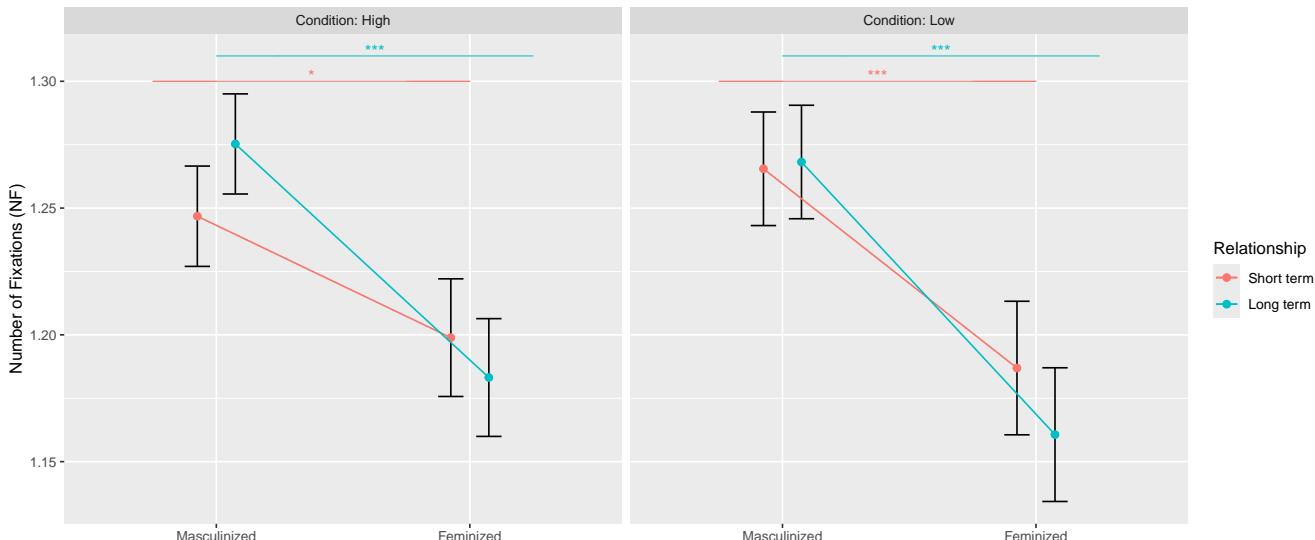
Condition	Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
							Difference	SE	z	p
High	Short term	Masculinized	1.25	0.02	1.21	1.29	0.05	0.02	2.54	<b>0.0111</b>
		Feminized	1.20	0.02	1.15	1.24				
	Long term	Masculinized	1.28	0.02	1.24	1.31	0.09	0.02	4.89	<b>&lt; 0.0001</b>
		Feminized	1.18	0.02	1.14	1.23				
Low	Short term	Masculinized	1.27	0.02	1.22	1.31	0.08	0.02	3.67	<b>&lt; 0.001</b>
		Feminized	1.19	0.03	1.14	1.24				
	Long term	Masculinized	1.27	0.02	1.22	1.31	0.11	0.02	5.01	<b>&lt; 0.0001</b>
		Feminized	1.16	0.03	1.11	1.21				

Note: EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.3.6 Figure for the NF model

This figure summarizes the the NF model results.

```
pm3 <- plot.exp(model = mod3,
                  y.pos = c(1.3, 1.3, 1.31, 1.31),
                  y.lab = "Number of Fixations (NF)")
pm3
```



**Figure S20.** Differences in the Number of Fixations (NF) to masculinized and feminized stimuli, by relationship (short term, long term) and condition (low, high). Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S14 and S17). In all cases, significant effects are represented with lines and stars: \* $p < 0.05$ , \*\* $p < 0.01$ , \*\*\* $p < 0.001$ .

## 4.4 Proportion of Chosen Faces (PCF)

### 4.4.1 Data

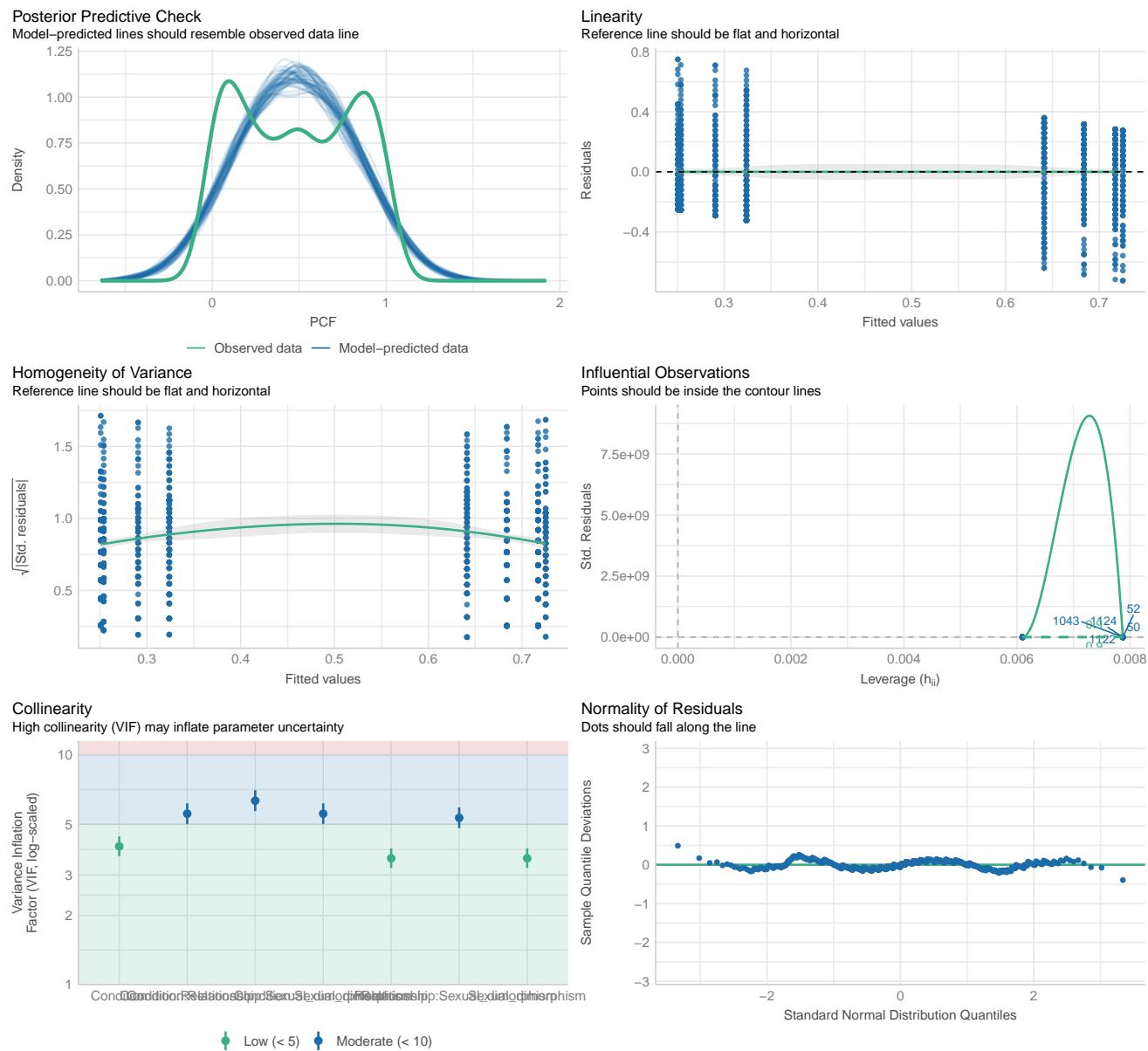
```
dat_m4 <- dat |>
  mutate(CF = as.numeric(recode(CF,
    "Yes" = "1",
    "No" = "0")))) |>
  group_by(ID, Stimulus, Relationship) |>
  filter(!(sum(CF[Sexual_dimorphism == "Masculinized"] == 0) > 0 &
    sum(CF[Sexual_dimorphism == "Feminized"] == 0) > 0)) |>
  ungroup() |>
  group_by(ID, Sexual_dimorphism, Relationship, Condition,
    Freq_partner_physical_violence,
    Freq_partner_sexual_violence,
    Freq_partner_infidelity,
    Men_perceived_as_dangerous,
    Perceived_home_safety) |>
  summarise(CF = sum(CF)) |>
  ungroup() |>
  mutate(PCF = CF/30) |>
  drop_na()
```

### 4.4.2 Fit linear mixed model

```
mod4 <- lm(PCF ~ Condition * Relationship * Sexual_dimorphism,
            data = dat_m4)
```

**4.4.2.1 Model assumptions** Model assumptions were checked using the `check_model` function from the `performance` package (Lüdecke et al., 2021).

```
check_model(mod4)
```



**Figure S21.** Model assumptions. Plots represent prediction check, linearity, homogeneity of variance, influential observations, collinearity, and normality of residuals, respectively.

#### 4.4.3 Table of fixed effects

```
tab_m4 <- lm.anova.tab(model = mod4)
tab_m4$kab
```

**Table S18.** ANOVA-type table of fixed effects for the PCF model

Fixed effect	F	df	p	$\omega_p^2$
Condition	0.05	1, 1156	0.82	0.00
Relationship	0.02	1, 1156	0.89	0.00
Sexual dimorphism	740.26	1, 1156	< 0.0001	0.39
Condition × Relationship	0.01	1, 1156	0.94	0.00
Condition × Sexual dimorphism	2.06	1, 1156	0.15	0.00
Relationship × Sexual dimorphism	15.29	1, 1156	< 0.0001	0.01
Condition × Relationship × Sexual dimorphism	1.13	1, 1156	0.29	0.00
<hr/>				
$R^2$				
Unadjusted = 0.4				
Adjusted = 0.39				

#### 4.4.4 Estimated marginal means and *post-hoc* contrasts of significant effects

**4.4.4.1 Main effect: Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli. All estimated marginal means and contrasts were calculated using the `emmeans` function from the `emmeans` package (Lenth, 2024).

```
main.eff.contr(model = mod4,
                 emm_contr = emmeans(mod4, pairwise ~ Sexual_dimorphism))
```

**Table S19.** Estimated marginal and contrast between Sexual dimorphism levels for the PCF model

Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
					Difference	SE	z	p
Masculinized	0.69	0.01	0.67	0.71	0.41	0.02	27.17	< 0.0001
Feminized	0.28	0.01	0.26	0.30				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

**4.4.4.2 Interaction: Relationship × Sexual dimorphism** Table of estimated marginal means and contrasts between masculinized and feminized stimuli by relationship. All estimated marginal means and contrasts were calculated using the `emmeans` function from the `emmeans` package (Lenth, 2024).

```
inter.contr(model = mod4,
            emm_contr = emmeans(mod4, pairwise ~ Sexual_dimorphism | Relationship))
```

**Table S20.** Estimated marginal and contrast between Sexual dimorphism levels by Relationship for the PCF model

Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
						Difference	SE	z	p
Short term	Masculinized	0.66	0.02	0.63	0.69	0.36	0.02	16.56	< 0.0001
	Feminized	0.31	0.02	0.28	0.34				
Long term	Masculinized	0.72	0.02	0.69	0.75	0.47	0.02	21.86	< 0.0001
	Feminized	0.25	0.02	0.22	0.28				

*Note:* EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.4.5 Full design contrasts

Table of estimated marginal means and contrasts between masculinized and feminized stimuli including the full experimental design. All estimated marginal means and contrasts were calculated using the `emmeans` function from the `emmeans` package (Lenth, 2024).

```
full.contr(model = mod4,
            emm_contr = emmeans(mod4, pairwise ~ Sexual_dimorphism | Relationship + Condition))
```

**Table S21.** Estimated marginal and contrast between masculinized and feminized stimuli by Relationship and Condition for the PCF model

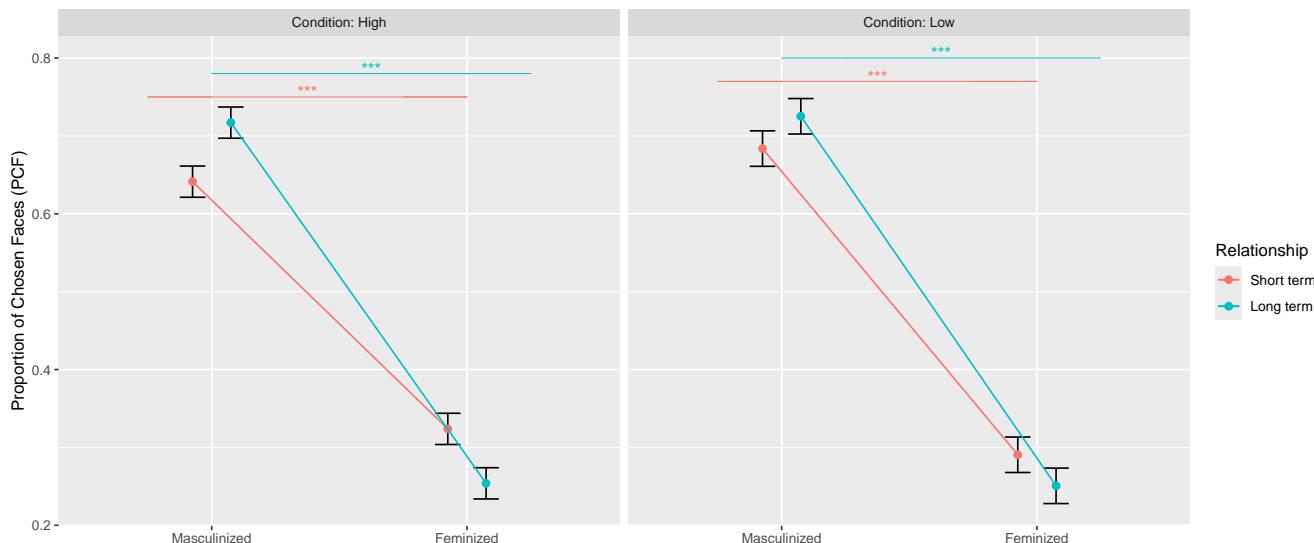
Condition	Relationship	Sexual dimorphism	EMM	SE	2.5%CI	97.5%CI	Contrasts (Masculinized - Feminized)			
							Difference	SE	z	p
High	Short term	Masculinized	0.64	0.02	0.60	0.68	0.32	0.03	11.20	< 0.0001
		Feminized	0.32	0.02	0.28	0.36				
	Long term	Masculinized	0.72	0.02	0.68	0.76	0.46	0.03	16.34	< 0.0001
		Feminized	0.25	0.02	0.21	0.29				
Low	Short term	Masculinized	0.68	0.02	0.64	0.73	0.39	0.03	12.21	< 0.0001
		Feminized	0.29	0.02	0.25	0.34				
	Long term	Masculinized	0.73	0.02	0.68	0.77	0.47	0.03	14.73	< 0.0001
		Feminized	0.25	0.02	0.21	0.30				

Note: EMM = estimated marginal mean. No degrees of freedom are reported, as an asymptotic method was used. Because of this, z rather than t scores are reported. Significant effects are in bold.

#### 4.4.6 Figure for the NF model

This figure summarizes the the NF model results.

```
pm4 <- plot.exp(model = mod4,
                  y.pos = c(0.75, 0.77, 0.78, 0.8),
                  y.lab = "Proportion of Chosen Faces (PCF)")
pm4
```



**Figure S22.** Differences in the proportion of masculinized and feminized chosen faces (PCF), by relationship (short term, long term) and condition (low, high). Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S18 and S21). In all cases, significant effects are represented with lines and stars: \*\*\* $p < 0.001$ .

## 5 Exploring interactions with covariates

In each case (i.e. for each dependent variable), we fitted five competing models, each one interacting with a different numeric variable related to the frequency of the current/last partner violent behaviors of perceived safety:

- Perception of men as dangerous
- Sexual violence
- Physical violence
- Infidelity
- Perceived home safety

Then, the model without covariates (fitted in section 4) and these five models with a covariate were compared using AIC scores (Akaike, 1974, 1998), and Akaike weights (see e.g., Portet, 2020; Wagenmakers & Farrell, 2004) to measure the relative probability of each compared model of being the *best* model within that set of compared models, for the collected data.

### 5.1 Duration of First Fixations (DFF)

#### 5.1.1 Fit linear models with interactions partner violence variables

```
# Model a: interaction with Men Perceived as Dangerous
mod1a <- lmer(DFF ~
  Condition * Relationship * Sexual_dimorphism * Men_perceived_as_dangerous +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m1)

# Model b: interaction with Frequency of Partner Physical Violence
mod1b <- lmer(DFF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_physical_violence +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m1)

# Model c: interaction with Frequency of Partner Sexual Violence
mod1c <- lmer(DFF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_sexual_violence +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m1)

# Model d: interaction with Frequency of Partner Infidelity
mod1d <- lmer(DFF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_infidelity +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m1)

# Model e: interaction with Perceived Home Safety
mod1e <- lmer(DFF ~
  Condition * Relationship * Sexual_dimorphism * Perceived_home_safety +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m1)
```

#### 5.1.2 Model comparison

```
comp_m1_cov <- mods.comp(base.model = "mod1")
comp_m1_cov$kab
```

**Table S22.** *AICc comparison of DFF models with different covariates*

Model (by covariate)	<i>AICc</i>	$\Delta(AICc)$	<i>w(AICc)</i>	<i>Prob.</i>
Physical violence	415,840.86	0.00	0.98	1 / 1
Men perceived as dangerous	415,849.22	8.36	0.0151	1 / 65
Sexual violence	415,863.44	22.57	< 0.0001	1 / 79,746
Infidelity	415,863.87	23.00	< 0.00001	1 / 98,948
(No covariate)	415,869.67	28.80	< 0.00001	1 / 1,797,626
Perceived home safety	415,869.88	29.01	< 0.00001	1 / 1,997,454

*Note:* Models are sorted according to their *AICc* values, increasingly, so that the best model (lowest *AICc*) is always at the top. *Delta(AICc)* is the *AICc* difference from each model to the best (top) model. *w(AICc)*, are Akaike weights, which represent the ‘weight of evidence’ (Portet, 2020) of each model for being the best model of the collection, given the data. Similarly, *Prob.* refers to the probability of each model of being the best model, in relation to the best-supported (top) model, expressed as a fraction. For the model with no covariates, *df* = 13; for all models with covariates, *df* = 21.

### 5.1.3 Best model

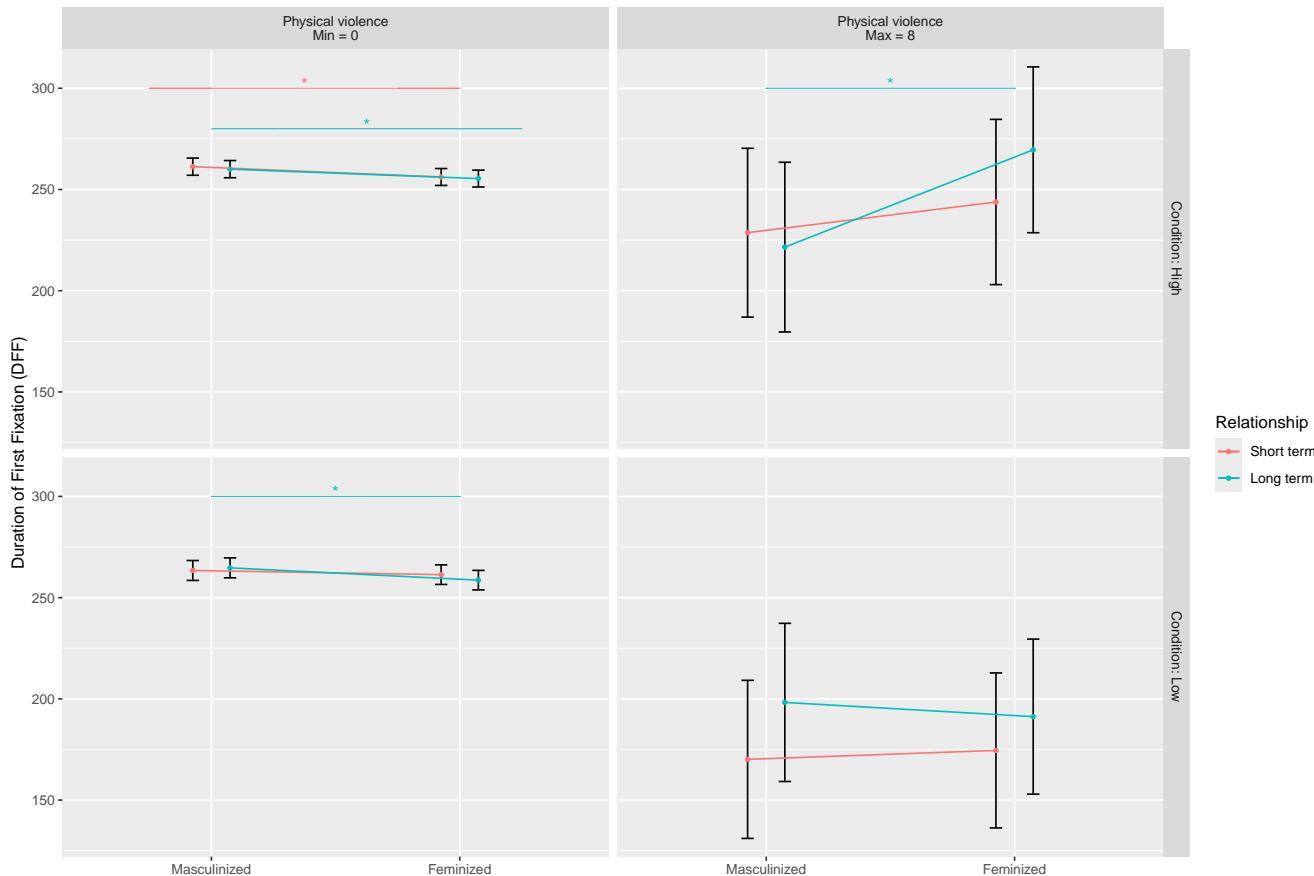
```
best_m1_cov <- eval(parse(text = rownames(data.frame(comp_m1_cov$comp))[1]))
tab_m1_cov <- lmer.anova.tab(model = best_m1_cov)
tab_m1_cov$kab
```

**Table S23.** *ANOVA-type table of fixed effects for the DFF model*

Fixed effect	<i>F</i>	<i>df</i>	<i>p</i>
Condition	0.41	1, 288.08	0.52
Relationship	0.48	1, 33366.71	0.49
Sexual dimorphism	12.37	1, 284.47	< 0.001
Physical violence	3.11	1, 286.78	0.08
Condition × Relationship	0.01	1, 33366.78	0.91
Condition × Sexual dimorphism	0.10	1, 284.46	0.75
Relationship × Sexual dimorphism	0.53	1, 33425.1	0.47
Condition × Physical violence	1.27	1, 286.78	0.26
Relationship × Physical violence	2.23	1, 33349.8	0.14
Sexual dimorphism × Physical violence	2.87	1, 289.65	0.09
Condition × Relationship × Sexual dimorphism	0.82	1, 33424.99	0.37
Condition × Relationship × Physical violence	0.33	1, 33349.7	0.57
Condition × Sexual dimorphism × Physical violence	2.12	1, 289.64	0.15
Relationship × Sexual dimorphism × Physical violence	0.31	1, 33366.93	0.58
Condition × Relationship × Sexual dimorphism × Physical violence	0.79	1, 33366.95	0.37
<b>Nakagawa’s <i>R</i><sup>2</sup></b>			
Conditional = 0.17			
Marginal = 0.0033			

#### 5.1.3.1 Table of fixed effects

```
pm1f <- plot.best.mod(best_model = best_m1_cov,
                      y.pos = c(300, NA, 280, 300, NA, NA, 300, NA),
                      y.lab = "Duration of First Fixation (DFF)")
```



**Figure S23.** Differences in the Duration of First Fixations (DFF) to masculinized and feminized stimuli, by relationship (short term, long term), condition (low, high), and the frequency of partner physical violence, which was the covariate included in the best-supported model. To illustrate the effect of the physical violence interaction, estimated marginal means are shown at minimum and maximum levels of this covariate. Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S22 and S23). In all cases, significant effects are represented with lines and stars: \* $p < 0.05$ .

### 5.1.3.2 Figure for the best model predicting DFF

## 5.2 Total Fixation Duration (TFD)

### 5.2.1 Fit linear models with interactions partner violence variables

```
# Model a: interaction with Men Perceived as Dangerous
mod2a <- lmer(TFD ~
  Condition * Relationship * Sexual_dimorphism * Men_perceived_as_dangerous +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m2)

# Model b: interaction with Frequency of Partner Physical Violence
mod2b <- lmer(TFD ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_physical_violence +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m2)

# Model c: interaction with Frequency of Partner Sexual Violence
mod2c <- lmer(TFD ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_sexual_violence +
```

```

(1 + Sexual_dimorphism | ID) + (1 | Stimulus),
data = dat_m2)

# Model d: interaction with Frequency of Partner Infidelity
mod2d <- lmer(TFD ~
    Condition * Relationship * Sexual_dimorphism * Freq_partner_infidelity +
    (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
    data = dat_m2)

# Model e: interaction with Perceived Home Safety
mod2e <- lmer(TFD ~
    Condition * Relationship * Sexual_dimorphism * Perceived_home_safety +
    (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
    data = dat_m2)

```

### 5.2.2 Model comparison

```

comp_m2_cov <- mods.comp(base.model = "mod2")
comp_m2_cov$kab

```

**Table S24.** *AICc comparison of TFD models with different covariates*

Model (by covariate)	AICc	$\Delta(AICc)$	w(AICc)	Prob.
Physical violence	260,523.3	0.00	1	1 / 1
Men perceived as dangerous	260,536.3	13.04	0.0015	1 / 677
Infidelity	260,540.2	16.88	< 0.001	1 / 4,637
Sexual violence	260,545.6	22.33	< 0.0001	1 / 70,620
Perceived home safety	260,552.2	28.91	< 0.00001	1 / 1,894,416
(No covariate)	260,575.4	52.08	< 0.00001	1 / 203,242,630,637

*Note:* Models are sorted according to their AICc values, increasingly, so that the best model (lowest AICc) is always at the top.  $\Delta(AICc)$  is the AICc difference from each model to the best (top) model. w(AICc), are Akaike weights, which represent the 'weight of evidence' (Portet, 2020) of each model for being the best model of the collection, given the data. Similarly, Prob. refers to the probability of each model of being the best model, in relation to the best-supported (top) model, expressed as a fraction. For the model with no covariates, df = 13; for all models with covariates, df = 21.

### 5.2.3 Best model

```

best_m2_cov <- eval(parse(text = rownames(data.frame(comp_m2_cov$comp))[1]))
tab_m2_cov <- lmer.anova.tab(model = best_m2_cov)
tab_m2_cov$kab

```

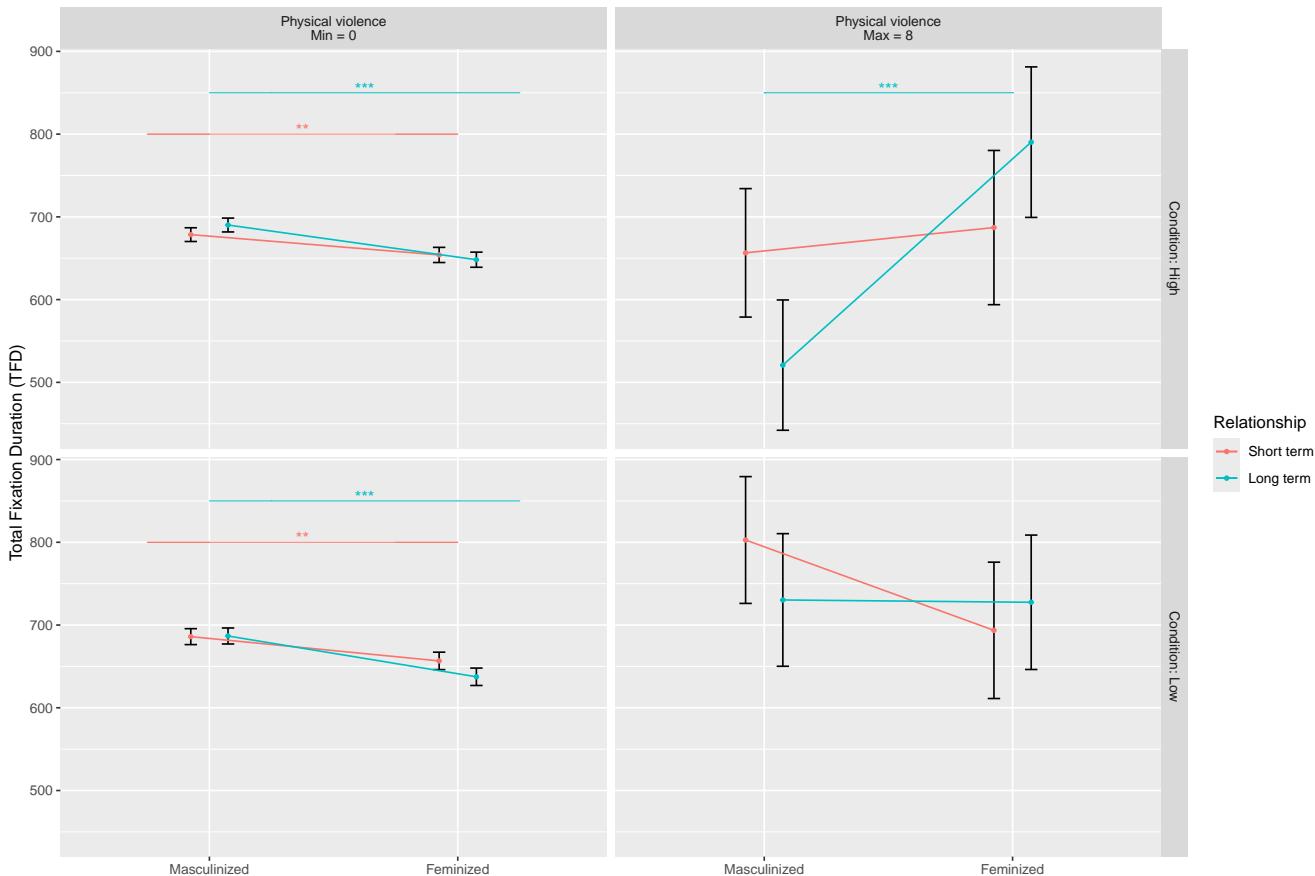
**Table S25.** ANOVA-type table of fixed effects for the TFD model

Fixed effect	F	df	p
Condition	0.01	1, 277.87	0.92
Relationship	0.98	1, 18829.35	0.32
Sexual dimorphism	48.48	1, 277.06	< <b>0.0001</b>
Physical violence	0.46	1, 282.81	0.5
Condition × Relationship	3.68	1, 18831	0.06
Condition × Sexual dimorphism	0.34	1, 273.45	0.56
Relationship × Sexual dimorphism	8.67	1, 18759.57	<b>0.0032</b>
Condition × Physical violence	0.57	1, 282.81	0.45
Relationship × Physical violence	0.23	1, 18898.14	0.63
Sexual dimorphism × Physical violence	2.95	1, 305.17	0.09
Condition × Relationship × Sexual dimorphism	0.04	1, 18761.5	0.84
Condition × Relationship × Physical violence	0.02	1, 18897.98	0.88
Condition × Sexual dimorphism × Physical violence	4.25	1, 305.16	<b>0.04</b>
Relationship × Sexual dimorphism × Physical violence	9.62	1, 18842.46	<b>0.0019</b>
Condition × Relationship × Sexual dimorphism × Physical violence	1.11	1, 18840.84	0.29
<b>Nakagawa's <math>R^2</math></b>			
Conditional = 0.17			
Marginal = 0.008			

### 5.2.3.1 Table of fixed effects

```
pm2f <- plot.best.mod(best_model = best_m2_cov,
                      y.pos = c(800, 800, 850, 850, NA, NA, 850, NA),
                      y.lab = "Total Fixation Duration (TFD)")

pm2f
```



**Figure S24.** Differences in the Total Fixation Duration (TFD) to masculinized and feminized stimuli, by relationship (short term, long term), condition (low, high), and the frequency of partner physical violence, which was the covariate included in the best-supported model. To illustrate the effect of the physical violence interaction, estimated marginal means are shown at minimum and maximum levels of this covariate. Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S24 and S25). In all cases, significant effects are represented with lines and stars: \*\* $p < 0.01$ , \*\*\* $p < 0.001$ .

### 5.2.3.2 Figure for the best model predicting TFD

## 5.3 Number of Fixations (NF)

### 5.3.1 Fit linear models with interactions partner violence variables

```
# Model a: interaction with Men Perceived as Dangerous
mod3a <- glmer(NF ~
  Condition * Relationship * Sexual_dimorphism * Men_perceived_as_dangerous +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m3,
  family = poisson)

# Model b: interaction with Frequency of Partner Physical Violence
mod3b <- glmer(NF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_physical_violence +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m3,
  family = poisson)

# Model c: interaction with Frequency of Partner Sexual Violence
```

```

mod3c <- glmer(NF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_sexual_violence +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m3,
  family = poisson)

# Model d: interaction with Frequency of Partner Infidelity
mod3d <- glmer(NF ~
  Condition * Relationship * Sexual_dimorphism * Freq_partner_infidelity +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m3,
  family = poisson)

# Model e: interaction with Perceived Home Safety
mod3e <- glmer(NF ~
  Condition * Relationship * Sexual_dimorphism * Perceived_home_safety +
  (1 + Sexual_dimorphism | ID) + (1 | Stimulus),
  data = dat_m3,
  family = poisson)

```

### 5.3.2 Model comparison

```

comp_m3_cov <- mods.comp(base.model = "mod3")
comp_m3_cov$kab

```

**Table S26.** *AICc comparison of NF models with different covariates*

Model (by covariate)	AICc	$\Delta(AICc)$	w(AICc)	Prob.
Physical violence	128,814.70	0.00	0.52	1 / 1.00
(No covariate)	128,816.11	1.41	0.26	1 / 2.02
Perceived home safety	128,816.58	1.87	0.2	1 / 2.55
Men perceived as dangerous	128,821.09	6.38	0.0212	1 / 24.32
Sexual violence	128,825.27	10.57	0.0026	1 / 197.34
Infidelity	128,826.33	11.62	0.0015	1 / 334.18

*Note:* Models are sorted according to their AICc values, increasingly, so that the best model (lowest AICc) is always at the top. *Delta(AICc)* is the AICc difference from each model to the best (top) model. *w(AICc)*, are Akaike weights, which represent the 'weight of evidence' (Portet, 2020) of each model for being the best model of the collection, given the data. Similarly, *Prob.* refers to the probability of each model of being the best model, in relation to the best-supported (top) model, expressed as a fraction. For the model with no covariates, *df* = 13; for all models with covariates, *df* = 21.

### 5.3.3 Best model

```

best_m3_cov <- eval(parse(text = rownames(data.frame(comp_m3_cov$comp))[1]))
tab_m3_cov <- glmer.anova.tab(model = best_m3_cov)
tab_m3_cov$kab

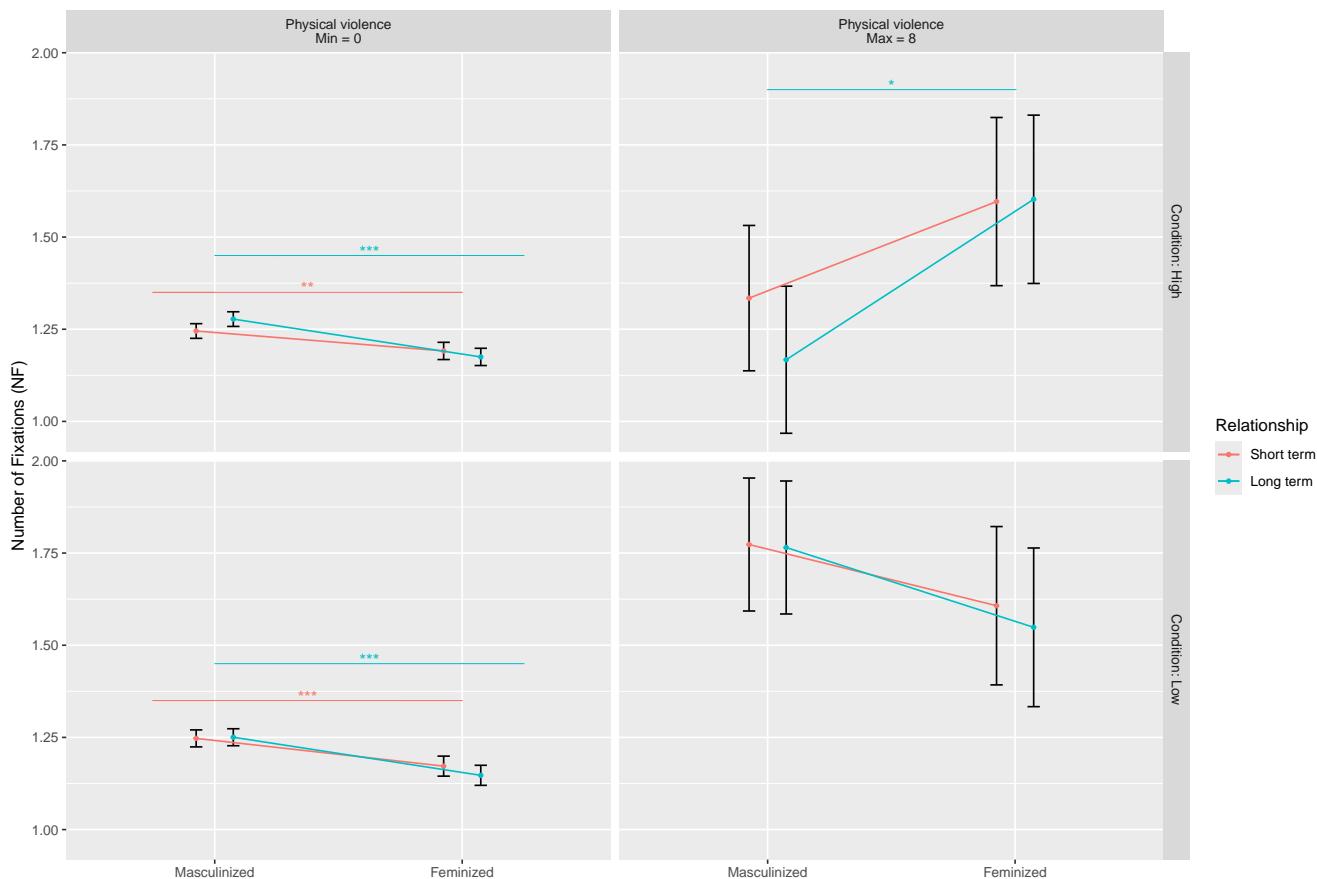
```

**Table S27.** ANOVA-type table of fixed effects for the NF model

Fixed effect	$\chi^2$	df	p
Condition	0.03	1	0.86
Relationship	0.03	1	0.87
Sexual dimorphism	38.93	1	< <b>0.0001</b>
Physical violence	6.05	1	<b>0.0139</b>
Condition × Relationship	2.26	1	0.13
Condition × Sexual dimorphism	0.95	1	0.33
Relationship × Sexual dimorphism	10.67	1	<b>0.0011</b>
Condition × Physical violence	2.36	1	0.12
Relationship × Physical violence	0.73	1	0.39
Sexual dimorphism × Physical violence	1.43	1	0.23
Condition × Relationship × Sexual dimorphism	0.33	1	0.57
Condition × Relationship × Physical violence	0.35	1	0.56
Condition × Sexual dimorphism × Physical violence	4.83	1	<b>0.0279</b>
Relationship × Sexual dimorphism × Physical violence	0.65	1	0.42
Condition × Relationship × Sexual dimorphism × Physical violence	1.39	1	0.24
<b>Nakagawa's <math>R^2</math></b>			
Conditional = 0.21			
Marginal = 0.011			

### 5.3.3.1 Table of fixed effects

```
pm3f <- plot.best.mod(best_model = best_m3_cov,
                      y.pos = c(1.35, 1.35, 1.45, 1.45, NA, NA, 1.9, NA),
                      y.lab = "Number of Fixations (NF)")
```



**Figure S25.** Differences in the Number of Fixations (NF) to masculinized and feminized stimuli, by relationship (short term, long term), condition (low, high). In this case, the best model was the model without covariates. Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S26 and S27). In all cases, significant effects are represented with lines and stars: \* $p < 0.05$ , \*\* $p < 0.01$ , \*\*\* $p < 0.001$ .

### 5.3.3.2 Figure for the best model predicting NF

## 5.4 Proportion of Chosen Faces (PCF)

### 5.4.1 Fit linear models with interactions partner violence variables

```
# Model a: interaction with Men Perceived as Dangerous
mod4a <- lm(PCF ~
             Condition * Relationship * Sexual_dimorphism * Men_perceived_as_dangerous,
             data = dat_m4)

# Model b: interaction with Frequency of Partner Physical Violence
mod4b <- lm(PCF ~
             Condition * Relationship * Sexual_dimorphism * Freq_partner_physical_violence,
             data = dat_m4)

# Model c: interaction with Frequency of Partner Sexual Violence
mod4c <- lm(PCF ~
             Condition * Relationship * Sexual_dimorphism * Freq_partner_sexual_violence,
             data = dat_m4)

# Model d: interaction with Frequency of Partner Infidelity
mod4d <- lm(PCF ~
```

```

Condition * Relationship * Sexual_dimorphism * Freq_partner_infidelity,
data = dat_m4)

# Model e: interaction with Perceived Home Safety
mod4e <- lm(PCF ~
    Condition * Relationship * Sexual_dimorphism * Perceived_home_safety,
    data = dat_m4)

```

#### 5.4.2 Model comparison

```

comp_m4_cov <- mods.comp(base.model = "mod4")
comp_m4_cov$kab

```

**Table S28.** *AICc comparison of PCF models with different covariates*

Model (by covariate)	<i>AICc</i>	$\Delta(AICc)$	<i>w(AICc)</i>	<i>Prob.</i>
Physical violence	139.95	0.00	0.97	1 / 1
(No covariate)	147.19	7.24	0.026	1 / 37
Perceived home safety	153.89	13.93	< 0.001	1 / 1,061
Men perceived as dangerous	154.22	14.26	< 0.001	1 / 1,251
Infidelity	156.61	16.66	< 0.001	1 / 4,147
Sexual violence	163.12	23.17	< 0.00001	1 / 107,471

*Note:* Models are sorted according to their *AICc* values, increasingly, so that the best model (lowest *AICc*) is always at the top. *Delta(AICc)* is the *AICc* difference from each model to the best (top) model. *w(AICc)*, are Akaike weights, which represent the 'weight of evidence' (Portet, 2020) of each model for being the best model of the collection, given the data. Similarly, *Prob.* refers to the probability of each model of being the best model, in relation to the best-supported (top) model, expressed as a fraction. For the model with no covariates, *df* = 13; for all models with covariates, *df* = 21.

#### 5.4.3 Best model

```

best_m4_cov <- eval(parse(text = rownames(data.frame(comp_m4_cov$comp))[1]))
tab_m4_cov <- lm.anova.tab(model = best_m4_cov)
tab_m4_cov$kab

```

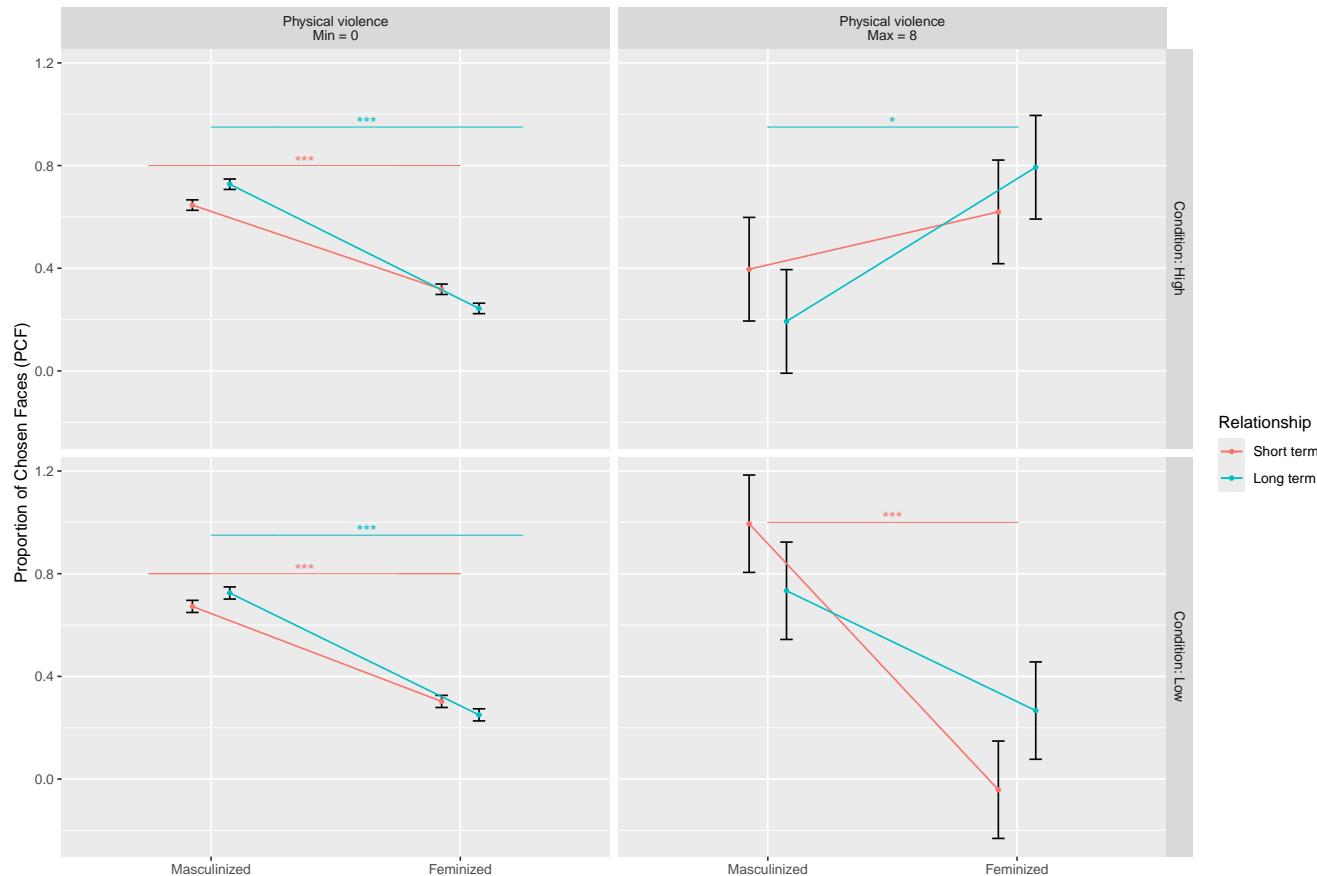
**Table S29.** ANOVA-type table of fixed effects for the PCF model

Fixed effect	<i>F</i>	<i>df</i>	<i>p</i>	$\omega_p^2$
Condition	0.06	1, 1148	0.81	0.00
Relationship	0.02	1, 1148	0.89	0.00
Sexual dimorphism	750.21	1, 1148	< <b>0.0001</b>	0.39
Physical violence	0.01	1, 1148	0.9	0.00
Condition × Relationship	0.01	1, 1148	0.94	0.00
Condition × Sexual dimorphism	2.09	1, 1148	0.15	0.00
Relationship × Sexual dimorphism	15.49	1, 1148	< <b>0.0001</b>	0.01
Condition × Physical violence	0.01	1, 1148	0.91	0.00
Relationship × Physical violence	0.00	1, 1148	0.98	0.00
Sexual dimorphism × Physical violence	2.35	1, 1148	0.13	0.00
Condition × Relationship × Sexual dimorphism	1.14	1, 1148	0.29	0.00
Condition × Relationship × Physical violence	0.02	1, 1148	0.88	0.00
Condition × Sexual dimorphism × Physical violence	16.45	1, 1148	< <b>0.0001</b>	0.01
Relationship × Sexual dimorphism × Physical violence	4.62	1, 1148	<b>0.0318</b>	0.00
Condition × Relationship × Sexual dimorphism × Physical violence	0.06	1, 1148	0.8	0.00
<i>R</i> <sup>2</sup>				
Unadjusted = 0.41				
Adjusted = 0.4				

#### 5.4.3.1 Table of fixed effects

```
pm4f <- plot.best.mod(best_model = best_m4_cov,
                      y.pos = c(0.8, 0.8, 0.95, 0.95, NA, 1, 0.95, NA),
                      y.lab = "Proportion of Chosen Faces (PCF)")

pm4f
```



**Figure S26.** Differences in the Proportion of Chosen Faces (PCF) to masculinized and feminized stimuli, by relationship (short term, long term), condition (low, high), and the frequency of partner physical violence, which was the covariate included in the best-supported model. To illustrate the effect of the physical violence interaction, estimated marginal means are shown at minimum and maximum levels of this covariate. Dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S28 and S29. In all cases, significant effects are represented with lines and stars:  $*p < 0.05$ .

#### 5.4.3.2 Figure for the best model predicting PCF

## 6 Final figures and tables (included in the main paper)

### 6.1 Table of fixed effects for experimental design models

```
tab_m1$tab |>
  left_join(tab_m2$tab, by = "Fixed effect") |>
  left_join(tab_m3$tab, by = "Fixed effect") |>
  left_join(tab_m4$tab, by = "Fixed effect") |>
  slice_head(n = 7) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 13)),
        col.names = c("Fixed effect",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$\chi^2$",
                     "$df$",
                     "$p$",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$\omega^2_p$"),
        linesep = "")
```

```

caption = "Pattern of gaze and choice of face by resource availability condition,
relationship context and sexual dimorphism",
escape = FALSE) |>
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
add_header_above(c(" " = 1,
"Duration of First Fixations (DFF)" = 3,
"Total Fixation Duration (TFD)" = 3,
"Number of Fixations (NF)" = 3,
"Proportion of Chosen Faces (PCF)" = 4)) |>
footnote(general = "For models predicting fixation measurements (DFF, TFD, and NF) results
are from linear mixed models (general, in the case of DFF, TFD, generalized with a poisson
distribution in the case of NF); for PCF, results are from a linear model without
random effects, and $\\omega^2_p$ effect sizes are reported for each term. For further
details, including full model effect sizes (Nakagawa's conditional and marginal $R^2$ for
linear mixed models, and unadjusted and adjusted $R^2$ for the PCF linear model),
see tables \\@ref(tab:tab-mod1), \\@ref(tab:tab-mod2), \\@ref(tab:tab-mod3) and
\\@ref(tab:tab-mod4) in the Supplementary Material. Significant effects are in bold.",
threeparttable = TRUE,
footnote_as_chunk = TRUE,
escape = FALSE)

```

**Table 1.** Pattern of gaze and choice of face by resource availability condition, relationship context and sexual dimorphism

Fixed effect	Duration of First Fixations (DFF)			Total Fixation Duration (TFD)			Number of Fixations (NF)			Proportion of Chosen Faces (PCF)			
	F	df	p	F	df	p	$\chi^2$	df	p	F	df	p	
Condition	0.06	1, 290	0.8	0.02	1, 279.87	0.88	0.00	1	1	0.05	1, 1156	0.82	0.00
Relationship	0.08	1, 33368.84	<b>0.77</b>	1.31	1, 18832.3	0.25	0.03	1	0.87	0.02	1, 1156	0.89	0.00
Sexual dimorphism	10.91	1, 285.13	<b>0.0011</b>	47.00	1, 279.46	< 0.0001	38.12	1	< 0.0001	740.26	1, 1156	< 0.0001	0.39
Condition × Relationship	0.14	1, 33369.04	0.71	3.71	1, 18834.01	0.05	2.56	1	0.11	0.01	1, 1156	0.94	0.00
Condition × Sexual dimorphism	0.01	1, 285.13	0.93	0.95	1, 275.61	0.33	0.78	1	0.38	2.06	1, 1156	0.15	0.00
Relationship × Sexual dimorphism	0.46	1, 33424.64	0.5	5.47	1, 18762.57	<b>0.0194</b>	10.75	1	<b>0.001</b>	15.29	1, 1156	< 0.0001	0.01
Condition × Relationship × Sexual dimorphism	1.25	1, 33424.51	0.26	0.05	1, 18764.97	0.82	0.44	1	0.51	1.13	1, 1156	0.29	0.00

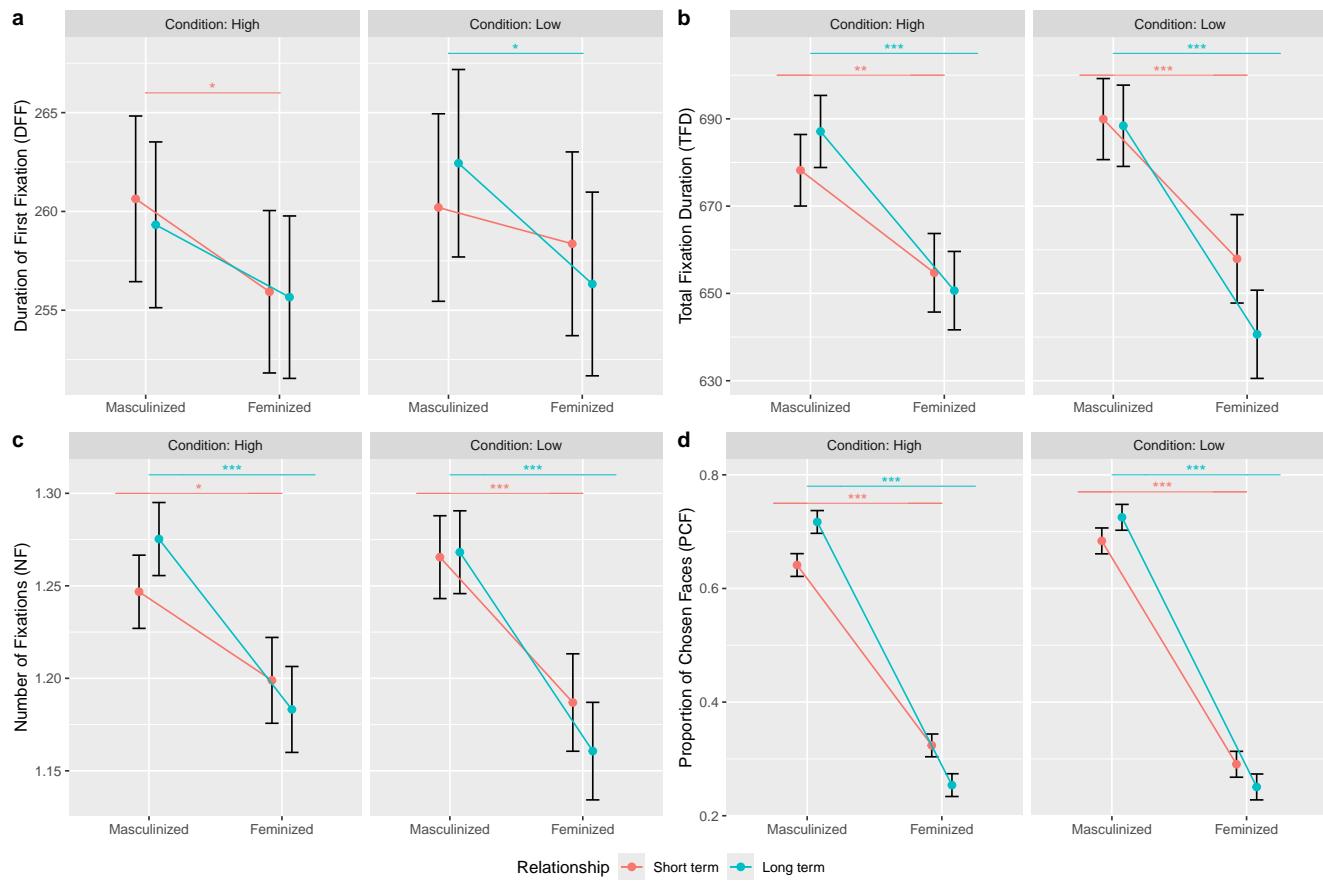
*Note:* For models predicting fixation measurements (DFF, TFD, and NF) results are from linear mixed models (general, in the case of DFF, TFD, generalized with a poisson distribution in the case of NF); for PCF, results are from a linear model without random effects, and  $\omega^2_p$  effect sizes are reported for each term. For further details, including full model effect sizes (Nakagawa's conditional and marginal  $R^2$  for linear mixed models, and unadjusted and adjusted  $R^2$  for the PCF linear model), see tables S7, S10, S14 and S18 in the Supplementary Material. Significant effects are in bold.

## 6.2 Figure of experimental design models

```

ggarrange(pm1, pm2, pm3, pm4,
common.legend = TRUE,
legend = "bottom",
labels = "auto")

```



**Figure 1.** Differences in responses to masculinized and feminized stimuli, by relationship (short term, long term), condition (low, high), and the frequency of partner physical violence, which was the covariate included in all best-supported models. To illustrate the effect of the physical violence interaction, estimated marginal means are shown at minimum and maximum levels of this covariate. **a.** Duration of First Fixations (DFF). **b.** Total Fixation Duration (TFD). **c.** Number of Fixations (NF). **d.** Proportion of Chosen Faces (PCF). Black dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S7 to S21 in the Supplementary Material. In all cases, significant effects are represented with lines and stars: \* $p < 0.05$ , \*\* $p < 0.01$ , \*\*\* $p < 0.001$ .

### 6.3 Table of fixed effects for best-supported models

```
tab_m1_cov$tab |>
  left_join(tab_m2_cov$tab, by = "Fixed effect") |>
  left_join(tab_m3_cov$tab, by = "Fixed effect") |>
  left_join(tab_m4_cov$tab, by = "Fixed effect") |>
  slice_head(n = 13) |>
  kable(digits = 2,
        booktabs = TRUE,
        align = c("l", rep("c", 13)),
        col.names = c("Fixed effect",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$\\chi^2$",
                     "$df$",
                     "$p$",
                     "$F$",
                     "$df$",
                     "$p$",
                     "$\\omega^2_p$"),
        linesep = "",
        caption = "Best-supported models predicting pattern of gaze and choice",
        escape = FALSE) |>
```

```

kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
add_header_above(c(" " = 1,
                  "Duration of First Fixations (DFF)" = 3,
                  "Total Fixation Duration (TFD)" = 3,
                  "Number of Fixations (NF)" = 3,
                  "Proportion of Chosen Faces (PCF)" = 4)) |>
footnote(general = "Best-supported model for predicting each dependent variable.  
For models predicting fixation measurements (DFF, TFD, and NF) results  
are from linear mixed models (general, in the case of DFF, TFD, generalized with a poisson  
distribution in the case of NF); for PCF, results are from a linear model without  
random effects, and $\\omega^2_p$ effect sizes are reported for each term. For further  
details, including full model effect sizes (Nakagawa's conditional and marginal $R^2$ for  
linear mixed models, and unadjusted and adjusted $R^2$ for the PCF linear model),  
see tables \\@ref(tab:comp-m1), \\@ref(tab:tab-best-mod2), \\@ref(tab:tab-best-mod3)  
and \\@ref(tab:tab-best-mod4) in the Supplementary Material.  
Significant effects are in bold.",  
threeparttable = TRUE,  
footnote_as_chunk = TRUE,  
escape = FALSE)

```

**Table 2.** Best-supported models predicting pattern of gaze and choice

Fixed effect	Duration of First Fixations (DFF)			Total Fixation Duration (TFD)			Number of Fixations (NF)			Proportion of Chosen Faces (PCF)			
	F	df	p	F	df	p	$\chi^2$	df	p	F	df	p	
Condition	0.41	1, 288.08	0.52	0.01	1, 277.87	0.92	0.03	1	0.86	0.06	1, 1148	0.81	0.00
Relationship	0.48	1, 33366.71	0.49	0.98	1, 18829.35	0.32	0.03	1	0.87	0.02	1, 1148	0.89	0.00
Sexual dimorphism	12.37	1, 284.47	< 0.001	48.48	1, 277.06	< 0.0001	38.93	1	< 0.0001	750.21	1, 1148	< 0.0001	0.39
Physical violence	3.11	1, 286.78	0.08	0.46	1, 282.81	0.5	6.05	1	<b>0.0139</b>	0.01	1, 1148	0.9	0.00
Condition × Relationship	0.01	1, 33366.78	0.91	3.68	1, 18831	0.06	2.26	1	0.13	0.01	1, 1148	0.94	0.00
Condition × Sexual dimorphism	0.10	1, 284.46	0.75	0.34	1, 273.45	0.56	0.95	1	0.33	2.09	1, 1148	0.15	0.00
Relationship × Sexual dimorphism	0.53	1, 33425.1	0.47	8.67	1, 18759.57	<b>0.0032</b>	10.67	1	<b>0.0011</b>	15.49	1, 1148	< 0.0001	0.01
Condition × Physical violence	1.27	1, 286.78	0.26	0.57	1, 282.81	0.45	2.36	1	0.12	0.01	1, 1148	0.91	0.00
Relationship × Physical violence	2.23	1, 33349.8	0.14	0.23	1, 18898.14	0.63	0.73	1	0.39	0.00	1, 1148	0.98	0.00
Sexual dimorphism × Physical violence	2.87	1, 289.65	0.09	2.95	1, 305.17	0.09	1.43	1	0.23	2.35	1, 1148	0.13	0.00
Condition × Relationship × Sexual dimorphism	0.82	1, 33424.99	0.37	0.04	1, 18761.5	0.84	0.33	1	0.57	1.14	1, 1148	0.29	0.00
Condition × Relationship × Physical violence	0.33	1, 33349.7	0.57	0.02	1, 18897.98	0.88	0.35	1	0.56	0.02	1, 1148	0.88	0.00
Condition × Sexual dimorphism × Physical violence	2.12	1, 289.64	0.15	4.25	1, 305.16	<b>0.04</b>	4.83	1	<b>0.0279</b>	16.45	1, 1148	< 0.0001	0.01

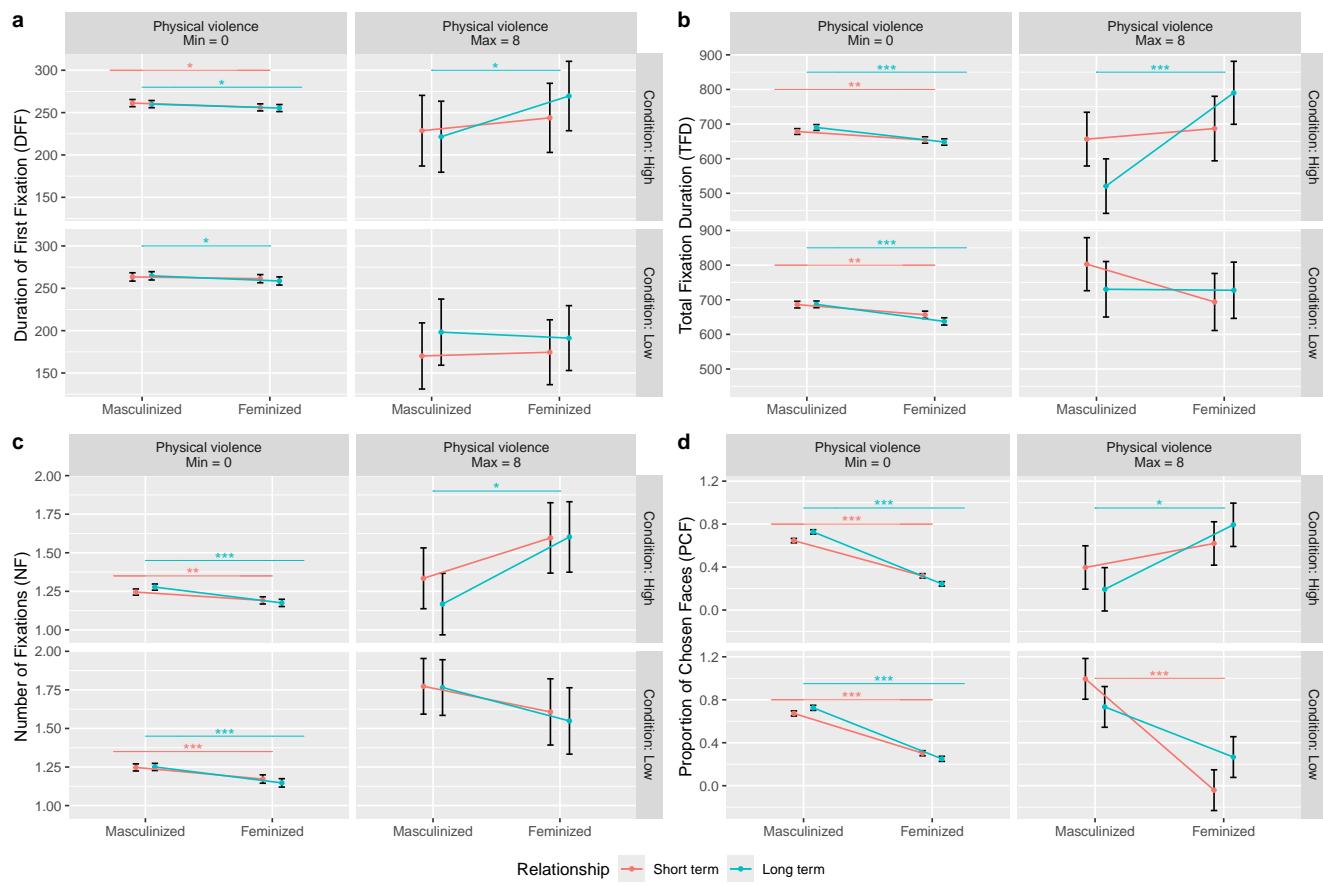
*Note:* Best-supported model for predicting each dependent variable. For models predicting fixation measurements (DFF, TFD, and NF) results are from linear mixed models (general, in the case of DFF, TFD, generalized with a poisson distribution in the case of NF); for PCF, results are from a linear model without random effects, and  $\omega^2_p$  effect sizes are reported for each term. For further details, including full model effect sizes (Nakagawa's conditional and marginal  $R^2$  for linear mixed models, and unadjusted and adjusted  $R^2$  for the PCF linear model), see tables S22, S25, S27 and S29 in the Supplementary Material. Significant effects are in bold.

## 6.4 Figure of experimental design models

```

ggarrange(pm1f, pm2f, pm3f, pm4f,
          common.legend = TRUE,
          legend = "bottom",
          labels = "auto")

```



**Figure 2.** Differences in responses to masculinized and feminized stimuli, by relationship (short term, long term) and condition (low, high). **a.** Duration of First Fixations (DFF). **b.** Total Fixation Duration (TFD). **c.** Number of Fixations (NF). **d.** Proportion of Chosen Faces (PCF). Black dots and bars represent estimated marginal means  $\pm$  standard errors. For detailed results, see Tables S7 to S29 in the Supplementary Material. In all cases, significant effects are represented with lines and stars:  $*p < 0.05$ ,  $**p < 0.01$ ,  $***p < 0.001$ .

## 7 Session info (for reproducibility)

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

R version 4.4.1 (2024-06-14)

Platform: x86\_64-pc-linux-gnu

attached base packages: stats4, stats, graphics, grDevices, utils, datasets, methods and base

other attached packages: pander(v.0.6.5), Hmisc(v.5.1-3), insight(v.0.20.4), effectsize(v.0.8.9), bbmle(v.1.0.25.1), gtools(v.3.9.5), FactoMineR(v.2.11), factoextra(v.1.0.7), scales(v.1.3.0), GGally(v.2.2.1), performance(v.0.12.3), kableExtra(v.1.4.0), emmeans(v.1.10.3), lmerTest(v.3.1-3), lme4(v.1.1-35.5), Matrix(v.1.7-0), readxl(v.1.4.3), ggpubr(v.0.6.0), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.1), dplyr(v.1.1.4), purrr(v.1.0.2), readr(v.2.1.5), tidyverse(v.1.3.1), tibble(v.3.2.1), ggplot2(v.3.5.1), tidyverse(v.2.0.0), ggstats(v.0.6.0), MASS(v.7.3-61), car(v.3.1-2), carData(v.3.0-5) and knitr(v.1.48)

loaded via a namespace (and not attached): RColorBrewer(v.1.1-3), rstudioapi(v.0.16.0), datawizard(v.0.12.3), magrittr(v.2.0.3), TH.data(v.1.1-2), estimability(v.1.5.1), farver(v.2.1.2), nloptr(v.2.1.1), rmarkdown(v.2.28), vctrs(v.0.6.5), minqa(v.1.2.7), base64enc(v.0.1-3), rstatix(v.0.7.2), htmltools(v.0.5.8.1), broom(v.1.0.6), cellranger(v.1.1.0), Formula(v.1.2-5), htmlwidgets(v.1.6.4), pbkrtest(v.0.5.3), plyr(v.1.8.9), sandwich(v.3.1-1), zoo(v.1.8-12), lifecycle(v.1.0.4), pkgconfig(v.2.0.3), R6(v.2.5.1), fastmap(v.1.2.0), di-

*gest(v.0.6.37), numDeriv(v.2016.8-1.1), colorspace(v.2.1-1), patchwork(v.1.3.0), labeling(v.0.4.3), fansi(v.1.0.6), timechange(v.0.3.0), abind(v.1.4-5), mgcv(v.1.9-1), compiler(v.4.4.1), withr(v.3.0.1), htmlTable(v.2.4.3), backports(v.1.5.0), highr(v.0.11), ggsignif(v.0.6.4), scatterplot3d(v.0.3-44), flashClust(v.1.01-2), tools(v.4.4.1), foreign(v.0.8-86), nnet(v.7.3-19), glue(v.1.7.0), nlme(v.3.1-165), grid(v.4.4.1), checkmate(v.2.3.2), cluster(v.2.1.6), see(v.0.8.5), generics(v.0.1.3), gtable(v.0.3.5), tzdb(v.0.4.0), data.table(v.1.15.4), hms(v.1.1.3), xml2(v.1.3.6), utf8(v.1.2.4), ggrepel(v.0.9.5), pillar(v.1.9.0), splines(v.4.4.1), lattice(v.0.22-5), survival(v.3.7-0), tidyselect(v.1.2.1), gridExtra(v.2.3), bookdown(v.0.40), svglite(v.2.1.3), xfun(v.0.47), DT(v.0.33), stringi(v.1.8.4), yaml(v.2.3.10), boot(v.1.3-30), evaluate(v.0.24.0), codetools(v.0.2-20), multcompView(v.0.1-10), cli(v.3.6.3), rpart(v.4.1.23), xtable(v.1.8-4), parameters(v.0.22.2), systemfonts(v.1.1.0), munsell(v.0.5.1), Rcpp(v.1.0.13), coda(v.0.19-4.1), bdsmatrix(v.1.3-7), parallel(v.4.4.1), leaps(v.3.2), bayestestR(v.0.13.2), viridisLite(v.0.4.2), mvtnorm(v.1.2-5), rlang(v.1.1.4), cowplot(v.1.1.3) and multcomp(v.1.4-26)*

## 8 Supplementary references

- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723. <https://doi.org/10.1109/TAC.1974.1100705>
- Akaike, H. (1998). Information Theory and an Extension of the Maximum Likelihood Principle. In E. Parzen, K. Tanabe, & G. Kitagawa (Eds.), *Selected Papers of Hirotugu Akaike* (pp. 199–213). Springer. [https://doi.org/10.1007/978-1-4612-1694-0\\_15](https://doi.org/10.1007/978-1-4612-1694-0_15)
- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3), 255–278. <https://doi.org/10.1016/j.jml.2012.11.01>
- Kassambara, A., & Mundt, F. (2020). *Factoextra: Extract and visualize the results of multivariate data analyses* [R package version 1.0.7]. <https://CRAN.R-project.org/package=factoextra>
- Kroes, A. D. A., & Finley, J. R. (2023). Demystifying omega squared: Practical guidance for effect size in common analysis of variance designs. *Psychological Methods*. <https://doi.org/10.1037/met0000581>
- 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>
- Lê, S., Josse, J., & Husson, F. (2008). FactoMineR: A package for multivariate analysis. *Journal of Statistical Software*, 25(1), 1–18. <https://doi.org/10.18637/jss.v025.i01>
- Lenth, R. V. (2024). *Emmeans: Estimated marginal means, aka least-squares means* [R package version 1.10.3]. <https://CRAN.R-project.org/package=emmeans>
- Lüdecke, 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>
- Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining  $R^2$  from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133–142. <https://doi.org/10.1111/j.2041-210x.2012.00261.x>
- Portet, S. (2020). A primer on model selection using the Akaike Information Criterion. *Infectious Disease Modelling*, 5, 111–128. <https://doi.org/10.1016/j.idm.2019.12.010>
- Wagenmakers, E.-J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin & Review*, 11(1), 192–196. <https://doi.org/10.3758/BF03206482>
- 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 1.1.4]. <https://CRAN.R-project.org/package=dplyr>
- Wolen, A. R., Hartgerink, C. H., Hafen, R., Richards, B. G., Soderberg, C. K., & York, T. P. (2020). osfr: An R interface to the open science framework. *Journal of Open Source Software*, 5(46), 2071. <https://doi.org/10.21105/joss.02071>
- 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. (2020). *Kableextra: Construct complex table with 'kable' and pipe syntax* [R package version 1.3.1]. <https://CRAN.R-project.org/package=kableExtra>