Mental health and subjective well-being of trans and non-binary population in Colombia

Supplementary Material: Code and analyses

[masked for peer review][1,*]   [masked for peer review][1]   [masked for peer review][2]

[masked for peer review][3]   [masked for peer review][4]

27 May 2025

[1] [masked for peer review]
[2] [masked for peer review]
[3] [masked for peer review]
[4] [masked for peer review]

* Correspondence: [masked for peer review]

Description

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

[masked for peer review] (in prep). *Mental health and subjective well-being of trans and non-binary population in Colombia*

Data are available on the Open Science Framework (OSF): https://osf.io/n2ucs/?view_only=5e734d 6f8ecc4bfca278b61cb98f5cb9. The analyses were designed by [masked for peer review] and [masked for peer review] and performed by [masked for peer review]. This document and all its underlying code were created in R Markdown by [masked for peer review] using R and LaTeX.

Contents

---

1   Preliminaries

1.1 Load packages

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

Multi-model inference and model averaging was achieved using `MuMIn` (Bartoń, 2025), and model assumptions were performed using `performance` (Lüdecke et al., 2021).

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 6, at the end of the document.

```
library(ltm)
library(psych) # For statistical functions (e.g., Cronbach's alpha)
library(MuMIn) # For model selection and averaging
library(performance) # For model performance metrics
```

```r
library(readr) # For reading data files
library(scales) # For percent formatting
library(knitr)
library(kableExtra)
library(car)
library(tidyverse) # For data manipulation and piping
library(ggpubr)
library(gtsummary)
library(Hmisc)
library(magick)
library(ggplotify)
library(pdftools)
```

1.2 Custom functions

1.2.1 `pval.lev` and `pval.stars`

These functions take p-values and formats them, either in LaTeXand highlighting significant p-values in bold and representing all in an appropriate level, or as stars.

```r
# Function to format p-values for LaTeX output, highlighting significant values in bold
pval.lev <- function(pvals) {
  ifelse(pvals < 0.0001, "\\textbf{< 0.0001}", # Highlight very small p-values
    ifelse(pvals < 0.001, "\\textbf{< 0.001}", # Bold p-values < 0.001
      ifelse(pvals < 0.05, paste0("\\textbf{", round(pvals, 4), "}"), # Bold p-values < 0.05
        round(pvals, 2) # Round non-significant values to two decimal places
      )
    )
  )
}

# Function to add significance stars based on p-value thresholds
pval.stars <- function(pvals) {
  ifelse(pvals < 0.0001, "****", # Four stars for p < 0.0001
    ifelse(pvals < 0.001, "***", # Three stars for p < 0.001
      ifelse(pvals < 0.01, "**", # Two stars for p < 0.01
        ifelse(pvals < 0.05, "*", NA) # One star for p < 0.05, NA otherwise
      )
    )
  )
}
```

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).

```r
# Function to create a correlation matrix with significance levels in LaTeX format
corr.stars <- function(x) {
  require(Hmisc) # Load Hmisc package for correlation and p-value calculations
  x <- as.matrix(x) # Ensure input is a matrix
  R <- rcorr(x, type = "spearman")$r # Compute correlation coefficients
  p <- rcorr(x, type = "spearman")$P # Extract p-values for significance testing
  # Define symbols for significance levels, using LaTeX formatting for bold and stars
  mystars <- ifelse(p < .001, paste0("\\textbf{", round(R, 2), "***}"), # p < 0.001
    ifelse(p < .01, paste0("\\textbf{", round(R, 2), "**}"), # p < 0.01
      ifelse(p < .05, paste0("\\textbf{", round(R, 2), "*}"), # p < 0.05
        ifelse(p < .10, paste0(round(R, 2), "$^{\\dagger}$"), # p < 0.10 (trend level)
```

```
        format(round(R, 2), nsmall = 2) # Format non-significant values with two decimals
      )
    )
  )
)
# Construct a new matrix with correlation values and significance symbols
Rnew <- matrix(mystars, ncol = ncol(x))
# Ensure diagonal values remain the original correlation values (without significance symbols)
diag(Rnew) <- paste(diag(R), " ", sep = "")
# Assign row and column names for the formatted matrix
rownames(Rnew) <- colnames(x)
colnames(Rnew) <- paste(colnames(x), "", sep = "")
# Remove the upper triangle of the matrix (including the diagonal) for a clean presentation
Rnew <- as.matrix(Rnew)
Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
# Convert to a data frame for better handling and remove the last empty column
Rnew <- as.data.frame(Rnew)
Rnew <- cbind(Rnew[1:length(Rnew) - 1])
return(Rnew) # Return formatted correlation table
}
```

1.2.3 `dredge.plot`

This function generates a model selection plot from a `MuMIn::dredge()` model selection table. It visualizes the top 100 candidate models, showing which terms are included in each model and their relative importance based on Akaike weights. The function also dynamically creates a caption summarizing the number of fitted models, suitable for use as a LaTeX or R Markdown fig.cap. Additionally, it saves the plot as a PDF file in a PDF_plots/ subdirectory (relative to the project root), with the filename based on the input object's name

```
dredge.plot <- function(mod.sel.table) {
  # Ensure the input is a dredge object
  if (!inherits(mod.sel.table, "model.selection")) {
    stop("Input must be a MuMIn::dredge() model selection table.")
  }
  # Extract the first model call from the dredge object
  model_call <- attr(mod.sel.table, "model.calls")[[1]]
  # Extract the response variable from the formula
  response_var <- all.vars(model_call$formula)[1]
  # Apply replacement logic
  response_label <- gsub("_", " ", response_var)
  # Create output directory if it doesn't exist
  dir.create("PDF_plots", showWarnings = FALSE)
  # Define output file path
  file_path <- file.path("PDF_plots", paste0(response_var, ".pdf"))
  # Save the plot to PDF
  pdf(file_path, width = 8, height = 6)
  plot_object <- plot(mod.sel.table[1:100, ], col = "#A92395FF",
                      main = NULL)
  dev.off()
  # Display the plot in the active graphics device (console/Rmd)
  plot(mod.sel.table[1:100, ], col = "#A92395FF",
       main = NULL, par.vlab = NULL)
  # Generate the caption
  caption <- paste0(
    "Model selection plot. Each row represents one of the top 100 models ",
    "(ranked by $AICc$) out of ", formatC(nrow(mod.sel.table), big.mark = ","),
    " fitted models. Purple cells indicate included terms, with row height corresponding",
```

```
    " to each model's relative support based on its Akaike weight ($\\\\omega$)"
  )
  # Return both the plot object and the caption
  return(list(plot = plot_object, caption = caption))
}
```

1.2.4 `avg.mod.summary`

This function creates a formatted summary table for model-averaged coefficients obtained from `MuMIn::model.avg()`. It includes coefficient estimates and 95% confidence intervals, significance levels (formatted for LaTeX output), a dynamically generated caption including the dependent variable name, and a footnote specifying reference levels for categorical predictors.

```
avg.mod.summary <- function(avg_model, data) {
  # Function to extract relevant reference levels
  extract_relevant_reference_levels <- function(avg_model, data) {
    # Identify factor variables in the dataset
    factor_vars <- names(data)[sapply(data, is.factor)]
    # Get reference levels for each factor variable
    ref_levels <- sapply(factor_vars, function(var) levels(data[[var]])[1])
    # Extract model terms
    model_terms <- rownames(summary(avg_model)$coefmat.full)
    # Detect which model terms correspond to factor variables
    relevant_factors <-
      factor_vars[sapply(factor_vars,
                         function(var) any(grepl(paste0("^", var), model_terms)))]
    # Keep only reference levels for relevant factors
    relevant_refs <- ref_levels[names(ref_levels) %in% relevant_factors]
    # Format the output
    if (length(relevant_refs) > 0) {
      paste0(
        "Reference levels: ",
        paste0(names(relevant_refs),
               " = ", relevant_refs,
               collapse = "; "
        )
      )
    } else {
      "No factor variables in the average model"
    }
  }

  # Compute CIs
  cis <- confint(avg_model, full = TRUE) |>
    as.data.frame() |>
    mutate(across(everything(), round, 3)) |>
    unite(col = "CI", `2.5 %`:`97.5 %`, sep = " - ")

  # Full average summary table
  summ_table <- summary(avg_model)$coefmat.full |>
    as.data.frame() |>
    bind_cols(cis) |>
    rownames_to_column(var = "term") |>
    mutate(term = term |>
             str_replace_all("Gender", "Gender: ") |>
             str_replace_all("Housing", "Housing: ") |>
             str_replace_all("Ethnicity", "Ethnicity: ") |>
```

```
                str_replace_all("Education", "Education: ") |>
                str_replace_all("Discrimination", "Discrimination: ") |>
                str_replace_all("Self_efficacy", "Self-efficacy") |>
                str_replace_all("_", " ")) |>
    select(term, Estimate, CI, `z value`, `Pr(>|z|)`) |>
    mutate(`Pr(>|z|)` = pval.lev(`Pr(>|z|)`)) |>
    arrange(term)

  kable_tab <- kable(
    summ_table,
    digits = 3,
    booktabs = TRUE,
    linesep = "",
    align = c("l", rep("c", 4)),
    caption = paste0(
      "Coefficient estimates for the ",
      as.character(avg_model$formula)[2] |>
        str_replace_all("_", " "),
      " model."
    ),
    col.names = c("Term", "$B$", "$95 \\% CIs$", "$z$", "$p$"),
    escape = FALSE
  ) |>
    kable_styling(latex_options = "HOLD_position") |>
    footnote(
      general = paste0(
        "This table includes estimates based on a full model average. ",
        "The full model average ensures that all variables are included in every model, ",
        "with coefficients set to zero when absent. As a consequence, it acts as a shrinkage ",
        "estimator, making estimates more conservative. ",
        "$B$ represents unstandardized model coefficients. ",
        extract_relevant_reference_levels(avg_model, data), # Now using the internal function
        ". Significant effects are in bold."
      ),
      threeparttable = TRUE,
      footnote_as_chunk = TRUE,
      escape = FALSE
    )

  # Return all intermediate objects
  return(list(table = kable_tab,
              coefficients = summ_table,
              reference_levels = extract_relevant_reference_levels(avg_model, data)))
}
```

1.2.5 `avg.mod.plot`

This function visualizes model-averaged estimates from `MuMIn::model.avg()`, displaying point estimates with 95% confidence intervals, predictor importance (mapped to point size and colour), and a dynamically generated annotation for the number of averaged model.

```
avg.mod.plot <- function(avg_mod) {
  # Extract model summary and transform into a tidy format
  x <- summary(avg_mod)$coefmat.full |>
    as_tibble(rownames = "key") |> # Convert row names to a "key" column
    bind_cols(
      confint(avg_mod, full = TRUE) |> as_tibble(), # Add confidence intervals
```

```r
    summary(avg_mod)$coef.nmod |>
      as_tibble() |>
      # Gather number of models per term
      pivot_longer(cols = everything(), names_to = "model", values_to = "value")
  ) |>
  mutate(
    avmod = deparse(substitute(avg_mod)) |>
      factor(), # Store model name as a factor
    value = value / max(value, na.rm = TRUE), # Normalize 'value' column
    sig = pval.stars(`Pr(>|z|)`) |>
      str_replace("\\.", "†"), # Convert p-values into significance stars
    key = key |>
      str_replace_all("Gender", "Gender: ") |>
      str_replace_all("Housing", "Housing: ") |>
      str_replace_all("Ethnicity", "Ethnicity: ") |>
      str_replace_all("Education", "Education: ") |>
      str_replace_all("Discrimination", "Discrimination: ") |>
      str_replace_all("Self_efficacy", "Self-efficacy") |>
      str_replace_all("_", " ")
  )

x <- x |>
  mutate(key = factor(key, levels = as.character(unique(x$key))))

# Get the number of averaged models
nMods <- dim(avg_mod$msTable)[1]

# Create the plot
ggplot(x, aes(x = key, y = Estimate)) +
  # Add horizontal reference line at zero
  geom_hline(yintercept = 0, color = "grey") +
  # Add points sized and colored by importance
  geom_point(aes(size = value, color = value), alpha = 0.7) +
  # Add error bars (confidence intervals)
  geom_errorbar(aes(ymin = `2.5 %`, ymax = `97.5 %`),
                colour = "black", width = 0.1
  ) +
  # Add additional points for emphasis
  geom_point(size = 1) +
  # Apply theme and labels
  theme_bw() +
  labs(x = NULL, y = "Estimate") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
  # Scale for importance (value) size
  scale_size_continuous(
    range = c(2, 8),
    breaks = seq(0, 1, by = 0.2)
  ) +
  # Legends for size and color
  guides(
    size = guide_legend(title = "Importance"),
    color = guide_legend(title = "Importance")
  ) +
  # Ensure x-axis labels remain in the correct order
  scale_x_discrete(
    labels = levels(x$key),
```

```
      expand = c(0, 0.5)
    ) +
    # Use plasma color scale, reversed so darker colors represent higher values
    scale_colour_viridis_c(
      option = "plasma", direction = -1,
      breaks = seq(0, 1, by = 0.2)
    ) +
    # Add significance labels next to points
    geom_text(aes(label = sig), y = x$`97.5 %`, vjust = -0.4) +
    # Add annotation indicating the number of averaged models
    annotate(
      "text",
      x = Inf, y = Inf,
      label = paste("Models averaged = ", nMods),
      hjust = 1.1, vjust = 1.8,
      size = 4,
      fontface = "bold.italic"
    )
}
```

1.3   Load data

Load raw CSV data

```
data_RAW <- read_csv("data/data.csv")
```

1.3.1   Define PANAS Subscales (Positive & Negative Affect)

The Positive and Negative Affect Schedule (PANAS) is a widely used self-report questionnaire that measures two independent dimensions of affect:

- Positive Affect (PA): Reflects the extent to which a person feels enthusiastic, active, and alert.
- Negative Affect (NA): Represents distress and unpleasurable engagement, including emotions such as fear, anger, and nervousness.

The following lists define the PANAS subscales based on their respective items.

```
# List of PANAS Positive Affect (PANAS_P) items
PANAS_P <- c(
  "PANASB_1", "PANASB_3", "PANASB_5", "PANASB_9",
  "PANASB_10", "PANASB_12", "PANASB_14", "PANASB_16",
  "PANASB_17", "PANASB_19"
)

# List of PANAS Negative Affect (PANAS_N) items
PANAS_N <- c(
  "PANASB_2", "PANASB_4", "PANASB_6", "PANASB_7",
  "PANASB_8", "PANASB_11", "PANASB_13", "PANASB_15",
  "PANASB_18", "PANASB_20"
)
```

1.4   Internal consistency

1.4.1   Calculate Cronbach's Alpha for Different Scales

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

```r
# Compute Cronbach's alpha for the Self-efficacy (EAG) scale
alpha_EAG <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA (missing values)
  select(starts_with("EAG_")) |> # Select all columns starting with "EAG_"
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE) # Compute Cronbach's alpha

# Compute Cronbach's alpha for the Life Satisfaction (SWLS) scale
alpha_SWLS <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(starts_with("SWLS_")) |> # Select all columns starting with "SWLS_"
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for the Resilience (EBR) scale
alpha_EBR <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(starts_with("EBR_")) |> # Select all columns starting with "EBR_"
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for the Depression (EBD) scale (after recoding responses)
alpha_EBD <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(starts_with("EBD_")) |> # Select all columns starting with "EBD_"
  mutate(across(everything(), ~ ifelse(is.na(.x), NA, .x - 1))) |> # Adjust values
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for the Social Support (MOS2) scale
alpha_MOS2 <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(starts_with("MOS2_")) |> # Select all columns starting with "MOS2_"
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for PANAS Positive Affect (PANAS_P)
alpha_PANAS_P <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(all_of(PANAS_P)) |> # Select PANAS_P variables
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for PANAS Negative Affect (PANAS_N)
alpha_PANAS_N <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(all_of(PANAS_N)) |> # Select PANAS_N variables
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)

# Compute Cronbach's alpha for Community Cohesion (PCPS3) scale
alpha_PCPS3 <- data_RAW |>
  mutate(across(where(is.numeric), ~ na_if(., 99))) |> # Replace 99 with NA
  select(starts_with("PCPS3_")) |> # Select all columns starting with "PCPS3_"
  drop_na() |>
  cronbach.alpha(CI = TRUE, standardized = TRUE)
```

1.4.2   Table S1. Internal consistency of measured scales

The internal consistency of the measured scales was generally strong, with Cronbach's α values ranging from 0.767 to 0.977. In particular, the Social Support (MOS2) and Self-efficacy (EAG) scales exhibited excellent internal consistency, while the Depression (EBD) and Community Cohesion (PCPS3) scales had acceptable reliability, suggesting a slightly lower but still adequate level of internal consistency.

```r
tibble(
  Scale = c(
    "Self-efficacy$^1$",
    "Life Satisfaction$^2$",
    "Resilience$^3$",
    "Depression$^4$",
    "Social Support$^5$",
    "PANAS Positive$^6$",
    "PANAS Negative$^6$",
    "Community Cohesion$^7$"
  ),
  p = c(
    alpha_EAG$p, alpha_SWLS$p, alpha_EBR$p, alpha_EBD$p, alpha_MOS2$p,
    alpha_PANAS_P$p, alpha_PANAS_N$p, alpha_PCPS3$p
  ),
  n = c(
    alpha_EAG$n, alpha_SWLS$n, alpha_EBR$n, alpha_EBD$n, alpha_MOS2$n,
    alpha_PANAS_P$n, alpha_PANAS_N$n, alpha_PCPS3$n
  ),
  alpha = c(
    alpha_EAG$alpha, alpha_SWLS$alpha, alpha_EBR$alpha, alpha_EBD$alpha,
    alpha_MOS2$alpha, alpha_PANAS_P$alpha, alpha_PANAS_N$alpha, alpha_PCPS3$alpha
  ),
  ci2.5 = c(
    alpha_EAG$ci[1], alpha_SWLS$ci[1], alpha_EBR$ci[1], alpha_EBD$ci[1],
    alpha_MOS2$ci[1], alpha_PANAS_P$ci[1], alpha_PANAS_N$ci[1], alpha_PCPS3$ci[1]
  ),
  ci97.5 = c(
    alpha_EAG$ci[2], alpha_SWLS$ci[2], alpha_EBR$ci[2], alpha_EBD$ci[2],
    alpha_MOS2$ci[2], alpha_PANAS_P$ci[2], alpha_PANAS_N$ci[2], alpha_PCPS3$ci[2]
  )
) |>
  mutate(across(starts_with("ci"), round, 3)) |>
  unite(col = "CI", ci2.5:ci97.5, sep = " - ") |>
  kable(
    digits = 3,
    booktabs = TRUE,
    linesep = "",
    align = c("l", rep("c", 4)),
    caption = "Internal consistency of measured scales",
    col.names = c("Variable", "Items", "$n$", "$\\alpha$", "$95\\% CI$"),
    escape = FALSE
  ) |>
  kable_styling(latex_options = "HOLD_position") |>
  footnote(
    general = "95\\\\% confidence intervals were calculated with 1,000 bootstrap samples.
            Standardized Cronbach's alpha ($\\\\alpha$) coefficients were computed.
            $^1$\\\\cite{EAG}; $^2$\\\\cite{SWLS}; $^3$\\\\cite{EBR}; $^4$\\\\cite{EBD};
            $^5$\\\\cite{MOS}; $^6$\\\\cite{PANAS}; $^7$\\\\cite{PCPS3}.",
    threeparttable = TRUE, footnote_as_chunk = TRUE, escape = FALSE
```

```
)
```

Table S1. *Internal consistency of measured scales*

Variable	Items	n	α	$95\%CI$
Self-efficacy[1]	10	223	0.905	0.876 — 0.928
Life Satisfaction[2]	5	253	0.869	0.838 — 0.896
Resilience[3]	4	278	0.861	0.827 — 0.893
Depression[4]	7	223	0.767	0.709 — 0.809
Social Support[5]	19	195	0.977	0.971 — 0.982
PANAS Positive[6]	10	285	0.884	0.857 — 0.907
PANAS Negative[6]	10	282	0.827	0.792 — 0.867
Community Cohesion[7]	3	281	0.769	0.703 — 0.825

Note: 95% confidence intervals were calculated with 1,000 bootstrap samples. Standardized Cronbach's alpha (α) coefficients were computed. [1]Baessler and Schwarzer, 1996; [2]Diener et al., 1985; [3]Sinclair and Wallston, 2004; [4]Andresen et al., 1994; [5]Sherbourne and Stewart, 1991; [6]Watson et al., 1988; [7]Stanton et al., 2017.

2 Data Preprocessing

2.1 Renaming, recoding, and filtering

This section contains the code to load the raw dataset (`data_RAW`) and create a final dataset (`data`) for analysis. Steps include:

- Renaming variables to meaningful labels
- Handling missing values (99 → NA)
- Recoding categorical variables into more readable labels
- Estimating participant scores for item-based scales (Self-efficacy, Life Satisfaction, Resilience, Depression, Social Support, Positive Affect, Negative Affect, Community Cohesion)
- Filtering out ineligible cases (e.g., participants under 18)

To estimate participant scores for item-based scales (i.e., Self-efficacy, Life Satisfaction, Resilience, Depression, Social Support, Positive Affect, Negative Affect, and Community Cohesion), and to minimise missing data in multi-model inference procedures, we computed scores in accordance with recommended practices (see Graham, 2012), using Cronbach's alpha levels to guide acceptable thresholds for missing data. For scales with Cronbach's alpha values ≥ 0.90 (e.g., Self-efficacy and Social Support), scores were calculated when at least 50% of the items had been completed. For scales with alpha values $\geq 0.80 < 0.90$ (e.g., Life Satisfaction, Resilience, and PANAS), up to 20% of items could be missing. For scales with alpha values ≥ 0.70 and < 0.80 (e.g., Depression and Community Cohesion), scores were calculated only when participants had one or no missing item. Scores of participants who did not meet these criteria for a given scale were treated as missing data (i.e., their scale score was set to NA).

This procedure implies that the number of items used to compute scale scores may vary across participants. To account for this, all scale scores were calculated as the mean of available items, even for scales that are typically computed as a sum: Self-efficacy, Life Satisfaction, Resilience, Depression, and Social Support. For these scales, however, an alternative, more traditional sum score was also computed for participants with complete item responses, and this version was used to report more readable descriptive statistics in Table S3.

```
data <- data_RAW |>
  # Rename columns to meaningful names
  rename(
    Age = SD1,
    City = SD2,
    Gender = SD3,
    Sexualientation = SD4,
```

```
    Sex = SD5,
    Ethnicity = SD6,
    Farmer = SD7,
    Marital_Status = SD8,
    SES = SD9, # Socioeconomic Status
    Education = SD10,
    Children = SD11,
    Housing = SD12,
    Cohabitant = SD13,
    Monthly_Income = SD14,
    Income_Source = SD15,
    Employment = SD16,
    Job = SD17,
    # Disabilities and difficulties
    Hearing_Difficulties = SD18_1,
    Speaking_Difficulties = SD18_2,
    Seeing_Difficulties = SD18_3,
    Moving_Difficulties = SD18_4,
    Grabing_Difficulties = SD18_5,
    Understanding_Difficulties = SD18_6,
    Interacting_Difficulties = SD18_7,
    # Lifetime Prevalence (LP) of substance use
    LP_Alcohol = SD19_1_A,
    LP_Cigarette = SD19_2_A,
    LP_Cannabis = SD19_3_A,
    LP_Cocaine = SD19_4_A,
    LP_Basuco = SD19_5_A,
    LP_Inhalant = SD19_6_A,
    LP_Ecstasy = SD19_7_A,
    LP_Psilocybin = SD19_8_A,
    LP_LSD = SD19_9_A,
    LP_Tranquilizer = SD19_10_A,
    LP_Popper = SD19_11_A,
    LP_Anfetamines = SD19_12_A,
    LP_Heroine = SD19_13_A,
    # Last Month (LM) substance use
    LM_Alcohol = SD19_1_B,
    LM_Cigarette = SD19_2_B,
    LM_Cannabis = SD19_3_B,
    LM_Cocaine = SD19_4_B,
    LM_Basuco = SD19_5_B,
    LM_Inhalant = SD19_6_B,
    LM_Ecstasy = SD19_7_B,
    LM_Psilocybin = SD19_8_B,
    LM_LSD = SD19_9_B,
    LM_TRAN = SD19_10_B,
    LM_Popper = SD19_11_B,
    LM_Anfetamines = SD19_12_B,
    LM_Heroine = SD19_13_B,
    # Last Week (LW) substance use
    LW_Alcohol = SD19_1_C,
    LW_Cigarette = SD19_2_C,
    LW_Cannabis = SD19_3_C,
    LW_Cocaine = SD19_4_C,
    LW_Basuco = SD19_5_C,
    LW_Inhalant = SD19_6_C,
```

```r
  LW_Ecstasy = SD19_7_C,
  LW_Psilocybin = SD19_8_C,
  LW_LSD = SD19_9_C,
  LW_Tranquilizer = SD19_10_C,
  LW_Popper = SD19_11_C,
  LW_Anfetamines = SD19_12_C,
  LW_Heroine = SD19_13_C,
  Health = SD20_1,
  # Health and other variables
  Illness = SD21,
  Disease_Other = SD22_13_TEXT,
  PCPS1_4_Other = PCPS1_4_texto,
  eed1_7_Other = EED1_7_TEXT
) |>
# Replace character "99" with NA for missing values
mutate(across(where(is.character), ~ na_if(., "99"))) |>
# Replace numeric 99 with NA for missing values
mutate(across(where(is.numeric), ~ na_if(., 99))) |>
# Recode gender categories into descriptive labels
mutate(
  Gender = recode(
    Gender,
    "1" = "Male",
    "2" = "Female",
    "3" = "Androgynous",
    "4" = "Trans woman",
    "5" = "Trans man",
    "6" = "Trans feminine",
    "7" = "Trans masculine",
    "8" = "Queer",
    "9" = "Non-binary",
    "10" = "Don't know",
    "11" = "Other"
  )
) |>
# Create a broader Gender category for analysis
mutate(Gender = if_else(Gender %in% c(
  "Woman", "Trans feminine", "Transexual", "Travesti", "Trans woman"
),
"Trans woman",
if_else(Gender %in% c("Man", "Trans masculine", "Trans man"),
  "Trans man",
  "Non-binary"
)
)) |>
# Recode housing categories into descriptive labels
mutate(
  Housing = recode(
    Housing,
    "1" = "Home-owner",
    "2" = "Renting (entire home)",
    "3" = "Living with family",
    "4" = "Shared rental (room)",
    "5" = "Without permanent housing"
  )
) |>
```

```r
# Recode substance use responses from text to binary (1 = Yes, 0 = No)
mutate_at(
  c(
    "LP_Alcohol",
    "LP_Cigarette",
    "LP_Cannabis",
    "LP_Cocaine",
    "LP_Basuco",
    "LP_Inhalant",
    "LP_Ecstasy",
    "LP_Psilocybin",
    "LP_LSD",
    "LP_Tranquilizer",
    "LP_Popper",
    "LP_Anfetamines",
    "LP_Heroine",
    "LM_Alcohol",
    "LM_Cigarette",
    "LM_Cannabis",
    "LM_Cocaine",
    "LM_Basuco",
    "LM_Inhalant",
    "LM_Ecstasy",
    "LM_Psilocybin",
    "LM_LSD",
    "LM_TRAN",
    "LM_Popper",
    "LM_Anfetamines",
    "LM_Heroine",
    "LW_Alcohol",
    "LW_Cigarette",
    "LW_Cannabis",
    "LW_Cocaine",
    "LW_Basuco",
    "LW_Inhalant",
    "LW_Ecstasy",
    "LW_Psilocybin",
    "LW_LSD",
    "LW_Tranquilizer",
    "LW_Popper",
    "LW_Anfetamines",
    "LW_Heroine"
  ),
  ~ recode(.x, "1" = 1, "2" = 0)
) |>
# Select only relevant variables
select(
  -c(
    Codigo,
    ends_with("_TEXT"),
    Sexualientation,
    ends_with("_texto")
  )
) |>
# Recode ethnicity, farmer status, marital status, SES, and education
mutate(
```

```r
    Ethnicity = recode(
      Ethnicity,
      "1" = "Indigenous",
      "2" = "Rrom",
      "3" = "Afro-Colombian",
      "4" = "Afro-Colombian",
      "5" = "Afro-Colombian",
      "6" = "Afro-Colombian",
      "7" = "Afro-Colombian",
      "8" = "Afro-Colombian",
      "9" = "None"
    )
) |>
mutate(Farmer = recode(
  Farmer,
  "1" = "Yes",
  "2" = "No",
  "5" = NA_character_
)) |>
mutate(
  Marital_Status = recode(
    Marital_Status,
    "1" = "Married",
    "2" = "Single",
    "3" = "Widow/er",
    "4" = "Divorced",
    "5" = "Civil union",
    "6" = "Stable relationship"
  )
) |>
mutate(
  SES = recode_factor(
    SES,
    "1" = "Low",
    "2" = "Low",
    "7" = "Low",
    "3" = "Middle-low",
    "4" = "Middle-high",
    "5" = "High",
    "6" = "High"
  )
) |>
mutate(
  Education = recode_factor(
    Education,
    "1" = "No studies, illiterate",
    "2" = "No studies, literate",
    "3" = "Primary school (unfinished)",
    "4" = "Primary school",
    "5" = "Secondary school (unfinished)",
    "6" = "Secondary school",
    "7" = "Technical degree",
    "8" = "University (unfinished)",
    "9" = "University",
    "10" = "Postgraduate studies"
  )
```

```r
) |>
mutate(across(
  c(
    SD22_1:SD22_13,
    EED1_1,
    EED1_2,
    EED1_3,
    EED1_4,
    EED1_5,
    EED1_6,
    EED1_7,
    EED2_1:EED2_5
  ),
  ~ as.numeric(
    recode(
      as.character(.x),
      "1" = "1",
      "2" = "0",
      .default = NA_character_,
      .missing = NA_character_
    )
  )
)) |>
# Convert disability variables to binary (1 = Has difficulty, 0 = No difficulty)
mutate(across(
  ends_with("_Difficulties"),
  ~ case_when(.x == 99 ~ NA_real_, is.na(.x) ~ NA_real_, .x == 4 ~ 1, TRUE ~ 0)
)) |>
# Create a new variable 'difficulty_dichotomous' to indicate whether a person has
# any difficulties across different categories
mutate(Difficulty_Dichotomous = if_else(
  # If any of the difficulties variables (e.g., Hearing_Difficulties,
  # Speaking_Difficulties, etc.) are NA
  rowSums(across(ends_with("_Difficulties"), ~ is.na(.))) > 0,
  NA_real_, # Assign NA if any difficulty is missing
  # Otherwise, check if all seven difficulties are marked as '1' (indicating impairment)
  if_else(rowSums(across(ends_with("_Difficulties"), ~ . == 1)) == 7,
    1, # Assign 1 if the person has all seven difficulties
    0 # Assign 0 otherwise
  )
)) |>
# Recode PCPS1_1 to PCPS1_5: Convert 1 to 1 (yes) and 2 to 0 (no), with NA for other values
mutate(across(PCPS1_1:PCPS1_5, ~ case_when(
  . == 1 ~ 1, # Yes
  . == 2 ~ 0, # No
  TRUE ~ NA_real_ # Missing or other values
))) |>
# Recode PCPS2_1 to PCPS2_5: Convert 1 to 0 (no engagement), and 2-5 to 1 (some engagement)
mutate(across(PCPS2_1:PCPS2_5, ~ case_when(
  . == 1 ~ 0, # No engagement
  . %in% 2:5 ~ 1, # Some engagement
  TRUE ~ NA_real_ # Missing or other values
))) |>
mutate(across(starts_with("EBD_"), ~ ifelse(is.na(.x), NA, .x - 1))) |>
# Compute aggregate variables summarizing different aspects
mutate(
```

```r
    # Count the number of substances used in the last month
    Polyconsumption_Month = rowSums(across(LM_Alcohol:LM_Heroine, ~.), na.rm = TRUE),
    # Count the number of reported diseases or health conditions
    Disease_Burden = rowSums(across(SD22_1:SD22_13, ~.), na.rm = TRUE),
    # Count the number of group memberships (sum of binary indicators)
    Group_Membership = rowSums(across(PCPS1_1:PCPS1_5, ~.), na.rm = TRUE),
    # Count the number of community engagement activities
    Community_Engagement = rowSums(across(PCPS2_1:PCPS2_5, ~.), na.rm = TRUE),
    # The following line is commented out: it would sum discrimination experiences
    Discrimination = rowSums(across(EED1_1:EED1_7, ~.), na.rm = TRUE),
    Discrimination = ifelse(Discrimination >= 1, "Yes", "No"),
    Self_efficacy = if_else(rowSums(!is.na(across(starts_with("EAG_")))) >= 5,
      rowMeans(across(starts_with("EAG_")), na.rm = TRUE),
      NA_real_
    ),
    Life_Satisfaction = if_else(rowSums(!is.na(across(starts_with("SWLS_")))) >= 3,
      rowMeans(across(starts_with("SWLS_")), na.rm = TRUE),
      NA_real_
    ),
    Resilience = if_else(rowSums(!is.na(across(starts_with("EBR_")))) >= 3,
      rowMeans(across(starts_with("EBR_")), na.rm = TRUE),
      NA_real_
    ),
    Depression = if_else(rowSums(!is.na(across(starts_with("EBD_")))) >= 6,
      rowMeans(across(starts_with("EBD_")), na.rm = TRUE),
      NA_real_
    ),
    Social_Support = if_else(rowSums(!is.na(across(starts_with("MOS2_")))) >= 10,
      rowMeans(across(starts_with("MOS2_")), na.rm = TRUE),
      NA_real_
    ),
    PANAS_Positive_Affect = if_else(rowSums(!is.na(across(all_of(PANAS_P)))) >= 8,
      rowMeans(across(all_of(PANAS_P)), na.rm = TRUE),
      NA_real_
    ),
    PANAS_Negative_Affect = if_else(rowSums(!is.na(across(all_of(PANAS_N)))) >= 9,
      rowMeans(across(all_of(PANAS_N)), na.rm = TRUE),
      NA_real_
    ),
    Community_Cohesion = if_else(rowSums(!is.na(across(starts_with("PCPS3_")))) >= 2,
      rowMeans(across(starts_with("PCPS3_")), na.rm = TRUE),
      NA_real_
    ),
    # Sums of complete cases for descriptives
    Self_efficacy_sum = if_else(rowSums(!is.na(across(starts_with("EAG_")))) >= 1,
      rowSums(across(starts_with("EAG_")), na.rm = TRUE),
      NA_real_
    ),
    Life_Satisfaction_sum = if_else(rowSums(!is.na(across(starts_with("SWLS_")))) >= 1,
      rowSums(across(starts_with("SWLS_")), na.rm = TRUE),
      NA_real_
    ),
    Resilience_sum = if_else(rowSums(!is.na(across(starts_with("EBR_")))) >= 1,
      rowSums(across(starts_with("EBR_")), na.rm = TRUE),
      NA_real_
    ),
```

```r
    Depression_sum = if_else(rowSums(!is.na(across(starts_with("EBD_")))) >= 1,
      rowSums(across(starts_with("EBD_")), na.rm = TRUE),
      NA_real_
    ),
    Social_Support_sum = if_else(rowSums(!is.na(across(starts_with("MOS2_")))) >= 1,
      rowSums(across(starts_with("MOS2_")), na.rm = TRUE),
      NA_real_
    ),
) |>
# Select relevant variables
select(
  Age, Gender, Ethnicity, Marital_Status, SES, Education, Housing,
  Health, Polyconsumption_Month:Social_Support_sum
) |>
# Convert categorical variables Housing to Job into factors
mutate(Housing = as.factor(Housing)) |>
# Convert all remaining character variables to factors
mutate_if(is.character, as.factor) |>
filter(Age >= 18)
```

2.2   Missing Data

Create a summary of missing data for each variable in the final dataset

```r
Missing_data <- data |>
  select(-ends_with("_sum")) |>
  # Summarize across all columns, counting the number of NA values in each column
  summarise(across(everything(), ~ sum(is.na(.)))) |>
  # Convert the summary from wide format (one row, many columns) to long format
  pivot_longer(
    everything(), # Select all columns
    names_to = "Variable", # Store column names in a new variable "Variable"
    values_to = "NA_count" # Store the count of NAs in a new variable "NA_count"
  ) |>
  # Compute the proportion of missing values for each variable
  mutate(Proportion = NA_count / dim(data)[1]) # Divide NA count by total number of rows
```

2.2.1   Fig. S1. Proportion of missing data

To apply multi-model inference techniques such as dredge and model averaging, models must be fitted with complete data. Therefore, assessing the proportion of missing data per variable was crucial. While excessive missingness could lead to unreliable models, imputing missing values might reduce data credibility. Since no variable had an unacceptably high proportion of missing data, we opted not to impute missing values.

```r
Missing_data |>
  mutate_at("Variable", str_replace_all, "Self_efficacy", "Self-efficacy") |>
  mutate_at("Variable", str_replace_all, "_", " ") |>
  ggplot(aes(
    x = fct_reorder(Variable, Proportion, .desc = TRUE), # Reorder variables
    y = Proportion,
    fill = Proportion # Use fill color to indicate proportion of missing data
  )) +
  geom_col() + # Create bar plot
  # Add percentage labels on top of bars
  geom_text(aes(label = percent(Proportion, accuracy = 1)),
    vjust = -0.5, size = 2
  ) +
```

```
# Apply color gradient: Green (low missing data), Yellow (moderate), Red (high missing data)
scale_fill_viridis_c(
  option = "plasma", # Define color range
  direction = -1, # Reverse the color scale
  labels = percent_format(accuracy = 1), # Convert legend values to percentage format
  alpha = 0.7
) +
# Convert Y-axis (proportion of missing data) to a percentage scale
scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
# Add axis labels
labs(
  y = "Percentage of Missing Data", # Label for Y-axis
  x = "Variable" # Label for X-axis
) +
# Use a minimal theme for a cleaner visual appearance
theme_minimal() +
# Rotate X-axis labels for better readability
theme(axis.text.x = element_text(angle = 60, hjust = 1))
```
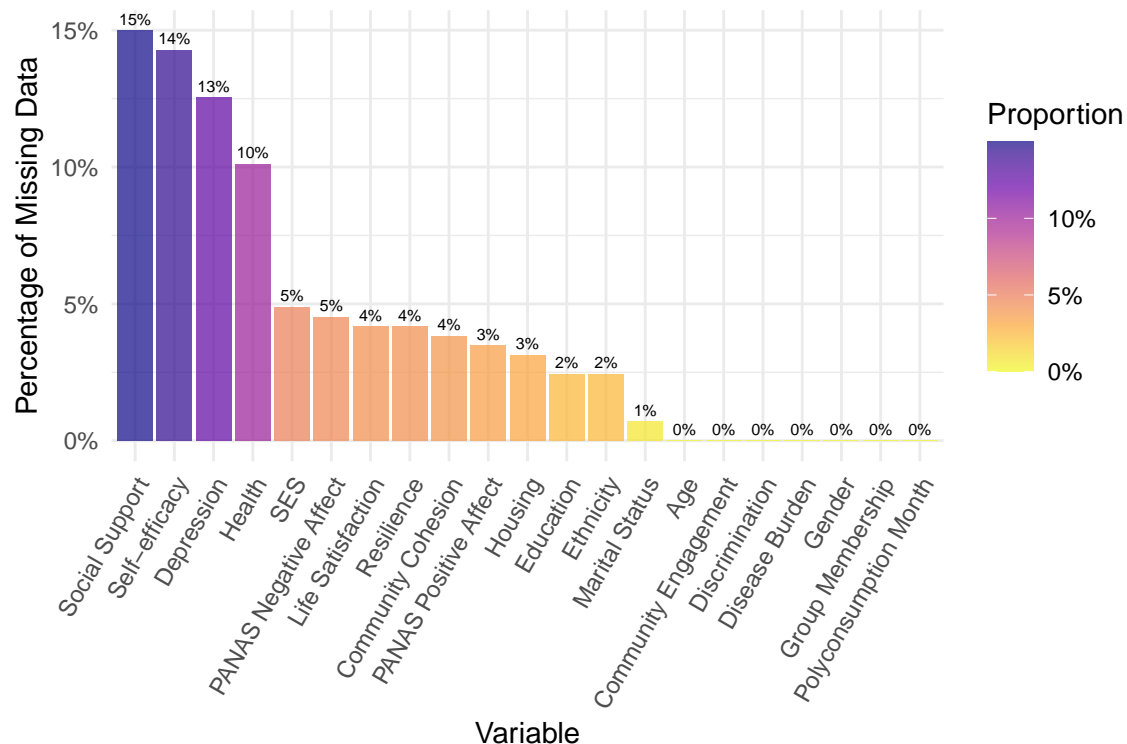

Figure S1. Proportion of missing data per variable. Variables are ordered from highest to lowest proportion of missing values. The color gradient indicates the proportion of missingness, with darker shades representing higher percentages.

3   Descriptives

3.1   Table S2. Socio-demographic characteristics by gender

```
data |>
  select(Age:Housing) |>
  rename_with(~ gsub("_", " ", .x)) |>
```

```r
tbl_summary(by = Gender,
            statistic = all_continuous() ~ "{mean}±{sd}",
            digits = all_continuous() ~ 2,
            missing_text = "Missing data") |>
add_n() |> # add column with total number of non-missing observations
bold_labels() |>
remove_footnote_header(columns = all_stat_cols()) |>
as_kable_extra(
  format = "latex",
  linesep = "",
  booktabs = TRUE,
  caption = "Sociodemographic characteristics of the study participants
            by gender identity"
) |>
kable_styling(latex_options = c("scale_down", "HOLD_position")) |>
footnote(
  general = "Age is reported as mean ± SD; all other variables are shown as frequency
            (\\\\%).",
  threeparttable = TRUE,
  footnote_as_chunk = TRUE,
  escape = FALSE
)
```

Table S2. *Sociodemographic characteristics of the study participants by gender identity*

Characteristic	N	Non-binary N = 70	Trans man N = 28	Trans woman N = 189
Age	287	30.80±10.57	28.96±8.36	36.61±11.75
Ethnicity	280			
Afro-Colombian		18 (26%)	10 (37%)	57 (31%)
Indigenous		5 (7.4%)	1 (3.7%)	17 (9.2%)
None		43 (63%)	15 (56%)	108 (58%)
Rrom		2 (2.9%)	1 (3.7%)	3 (1.6%)
Missing data		2	1	4
Marital Status	285			
Civil union		7 (10%)	6 (21%)	27 (14%)
Divorced		1 (1.4%)	0 (0%)	2 (1.1%)
Married		8 (11%)	4 (14%)	11 (5.9%)
Single		48 (69%)	13 (46%)	138 (74%)
Stable relationship		3 (4.3%)	4 (14%)	6 (3.2%)
Widow/er		3 (4.3%)	1 (3.6%)	3 (1.6%)
Missing data		0	0	2
SES	273			
Low		54 (81%)	20 (74%)	146 (82%)
Middle-low		12 (18%)	4 (15%)	29 (16%)
Middle-high		1 (1.5%)	2 (7.4%)	2 (1.1%)
High		0 (0%)	1 (3.7%)	2 (1.1%)
Missing data		3	1	10
Education	280			
No studies, illiterate		1 (1.5%)	1 (3.7%)	3 (1.6%)
No studies, literate		0 (0%)	0 (0%)	3 (1.6%)
Primary school (unfinished)		3 (4.4%)	0 (0%)	18 (9.7%)
Primary school		1 (1.5%)	0 (0%)	15 (8.1%)
Secondary school (unfinished)		12 (18%)	4 (15%)	54 (29%)
Secondary school		29 (43%)	9 (33%)	57 (31%)
Technical degree		13 (19%)	6 (22%)	23 (12%)
University (unfinished)		7 (10%)	5 (19%)	9 (4.9%)
University		1 (1.5%)	2 (7.4%)	3 (1.6%)
Postgraduate studies		1 (1.5%)	0 (0%)	0 (0%)
Missing data		2	1	4
Housing	278			
Home-owner		5 (7.4%)	2 (7.4%)	10 (5.5%)
Living with family		4 (5.9%)	3 (11%)	11 (6.0%)
Renting (entire home)		30 (44%)	17 (63%)	86 (47%)
Shared rental (room)		27 (40%)	4 (15%)	66 (36%)
Without permanent housing		2 (2.9%)	1 (3.7%)	10 (5.5%)
Missing data		2	1	6

Note: Age is reported as mean ± SD; all other variables are shown as frequency (%).

3.2   Table S3. Measured variables by gender

```
data |>
  select(
    Gender,
    Disease_Burden,
    Group_Membership,
    Community_Engagement,
    Discrimination,
```

```r
  Polyconsumption_Month,
  Self_efficacy_sum,
  Life_Satisfaction_sum,
  Resilience_sum,
  Depression_sum,
  Social_Support_sum,
  PANAS_Positive_Affect,
  PANAS_Negative_Affect,
  Community_Cohesion
) |>
rename_with(~ gsub("Self_efficacy", "Self-efficacy", .x)) |>
rename_with(~ gsub("_sum", " (sum)", .x)) |>
rename_with(~ gsub("_", " ", .x)) |>
tbl_summary(by = Gender,
            statistic = all_continuous() ~ "{mean} ± {sd}",
            missing_text = "Missing data") |>
add_n() |> # add column with total number of non-missing observations
bold_labels() |>
remove_footnote_header(columns = all_stat_cols()) |>
as_kable_extra(
  format = "latex",
  linesep = "",
  booktabs = TRUE,
  caption = "Other sociodemographic, health, and psychosocial characteristics
            of the study participants by gender identity"
) |>
kable_styling(latex_options = c("scale_down", "HOLD_position")) |>
footnote(
  general = "Continuous variables are reported as mean ± SD, and categorical variables
            as frequency (\\\\%). Missing data indicate the number of participants with
            insufficient responses to compute the corresponding score. For interpretability,
            Self-efficacy, Life Satisfaction, Resilience, Depression, and Social Support are
            shown as sum scores, calculated only for participants with complete item
            responses.",
  threeparttable = TRUE,
  footnote_as_chunk = TRUE,
  escape = FALSE
)
```

Table S3. *Other sociodemographic, health, and psychosocial characteristics of the study participants by gender identity*

Characteristic	N	Non-binary N = 70	Trans man N = 28	Trans woman N = 189
Disease Burden	287			
0		35 (50%)	11 (39%)	91 (48%)
1		20 (29%)	11 (39%)	58 (31%)
2		9 (13%)	3 (11%)	19 (10%)
3		2 (2.9%)	3 (11%)	8 (4.2%)
4		2 (2.9%)	0 (0%)	8 (4.2%)
5		1 (1.4%)	0 (0%)	3 (1.6%)
6		1 (1.4%)	0 (0%)	2 (1.1%)
Group Membership	287			
0		29 (41%)	10 (36%)	65 (34%)
1		32 (46%)	14 (50%)	84 (44%)
2		4 (5.7%)	2 (7.1%)	23 (12%)
3		1 (1.4%)	2 (7.1%)	9 (4.8%)
4		3 (4.3%)	0 (0%)	6 (3.2%)
5		1 (1.4%)	0 (0%)	2 (1.1%)
Community Engagement	287			
0		21 (30%)	4 (14%)	27 (14%)
1		9 (13%)	2 (7.1%)	43 (23%)
2		6 (8.6%)	6 (21%)	44 (23%)
3		11 (16%)	5 (18%)	39 (21%)
4		15 (21%)	8 (29%)	28 (15%)
5		8 (11%)	3 (11%)	8 (4.2%)
Discrimination	287	60 (86%)	21 (75%)	169 (89%)
Polyconsumption Month	287	2.16 ± 2.11	1.82 ± 2.02	2.37 ± 3.13
Self-efficacy (sum)	251	30 ± 8	26 ± 9	33 ± 7
Missing data		8	7	21
Life Satisfaction (sum)	284	21 ± 8	21 ± 9	23 ± 10
Missing data		1	0	2
Resilience (sum)	280	14.6 ± 5.0	12.5 ± 4.1	15.4 ± 4.6
Missing data		3	1	3
Depression (sum)	286	9.9 ± 4.4	9.7 ± 4.6	9.5 ± 4.2
Missing data		0	0	1
Social Support (sum)	250	60 ± 24	57 ± 22	60 ± 24
Missing data		8	7	22
PANAS Positive Affect	277	3.53 ± 1.09	3.31 ± 0.85	3.94 ± 0.91
Missing data		2	1	7
PANAS Negative Affect	274	2.31 ± 0.94	2.43 ± 0.89	2.50 ± 1.01
Missing data		2	2	9
Community Cohesion	276	4.15 ± 1.74	3.98 ± 1.44	4.68 ± 1.45
Missing data		5	0	6

Note: Continuous variables are reported as mean ± SD, and categorical variables as frequency (%). Missing data indicate the number of participants with insufficient responses to compute the corresponding score. For interpretability, Self-efficacy, Life Satisfaction, Resilience, Depression, and Social Support are shown as sum scores, calculated only for participants with complete item responses.

3.3 Figure S2. Distribution of participants' categorical and ordinal variables by gender

```r
# Define a custom colour palette using the 'plasma' viridis option,
# excluding the light yellow end by setting end = 0.75
custom_palette <- viridis::viridis(3, option = "plasma", begin = 0, end = 0.75)

p_desc1 <- data |>
  select(Gender,
         Ethnicity,
         Marital_Status,
         SES,
         Education,
         Housing,
         Disease_Burden,
         Group_Membership,
         Community_Engagement,
         Discrimination) |>
  mutate_all(as.factor) |>
  pivot_longer(
    cols = -Gender,
    names_to = "Variable", values_to = "Response"
  ) |>
  mutate(
    Response = as.factor(Response),
    Variable = str_wrap(Variable, width = 20)
  ) |>
  group_by(Gender, Variable, Response) |>
  summarise(n = n(), .groups = "drop") |>
  group_by(Gender, Variable) |>
  mutate(Percentage = n / sum(n) * 100,
         Variable = str_replace_all(Variable, "_", " ")) |>
  # Plot proportions instead of counts
  ggplot(aes(x = Response, y = Percentage, fill = Gender)) +
  geom_col(position = position_dodge(width = 0.9)) +
  facet_wrap(~Variable, scales = "free_x", ncol = 5) +
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +  # Show axis in %
  scale_fill_manual(values = custom_palette) +
  labs(x = NULL, y = "Percentage") +
  theme_minimal() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 52, hjust = 1))
p_desc1
```
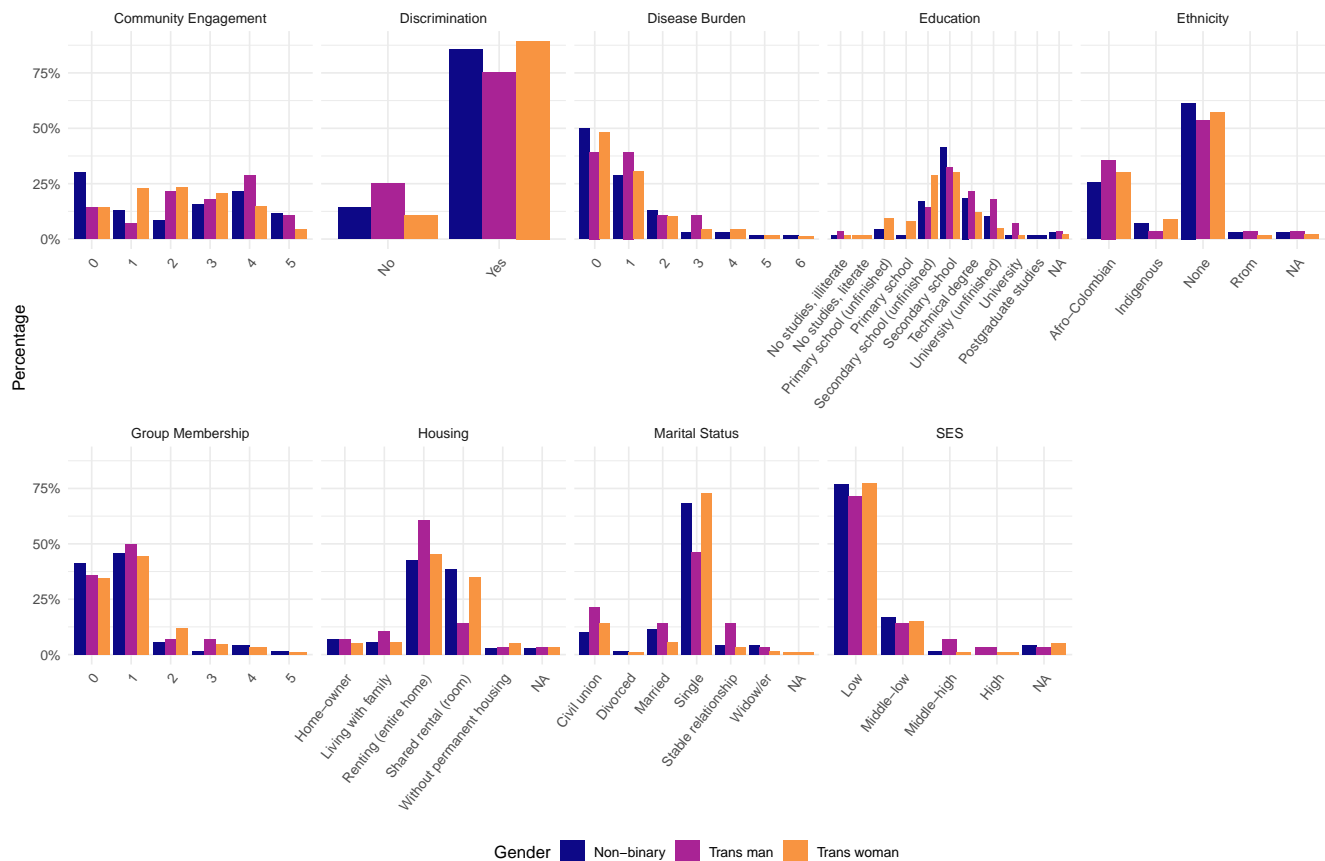
Figure S2. Distribution of categorical and ordinal variables by gender. Values represent the percentage of responses in each category. To facilitate comparison across genders, percentages were calculated within each gender group. Additional descriptive statistics are provided in Table S2.

3.4　Figure S3. Distribution of participants' measured numeric variables by gender

```r
# Transform the data
desc_long_data <- data |>
  select(
    Gender,
    Age,
    Polyconsumption_Month,
    Self_efficacy_sum,
    Life_Satisfaction_sum,
    Resilience_sum,
    Depression_sum,
    Social_Support_sum,
    PANAS_Positive_Affect,
    PANAS_Negative_Affect,
    Community_Cohesion
  ) |>
  pivot_longer(
    cols = -Gender,
    names_to = "Variable", values_to = "Value"
  ) |>
  mutate(
    Variable = str_replace_all(Variable, "Self_efficacy", "Self-efficacy"),
```

```
    Variable = str_replace_all(Variable, "_sum", " (sum)"),
    Variable = str_replace_all(Variable, "_", " "),
    Variable = str_wrap(Variable, width = 30)
  )

# Calculate group means for each Gender × Variable
desc_means <- desc_long_data |>
  group_by(Gender, Variable) |>
  summarise(Mean = mean(Value, na.rm = TRUE), .groups = "drop")

# Create the plot
p_desc2 <- ggplot(desc_long_data, aes(Value, fill = Gender, colour = Gender)) +
  geom_density(alpha = 0.6) +
  geom_vline(data = desc_means,
             aes(xintercept = Mean,
                 colour = Gender),
             linetype = "dashed", size = 0.8) +
  facet_wrap(~Variable, scales = "free", ncol = 5) +
  labs(y = "Density", x = NULL) +
  scale_fill_manual(values = custom_palette) +
  scale_colour_manual(values = custom_palette) +
  theme_minimal() +
  theme(legend.position = "bottom")
p_desc2
```
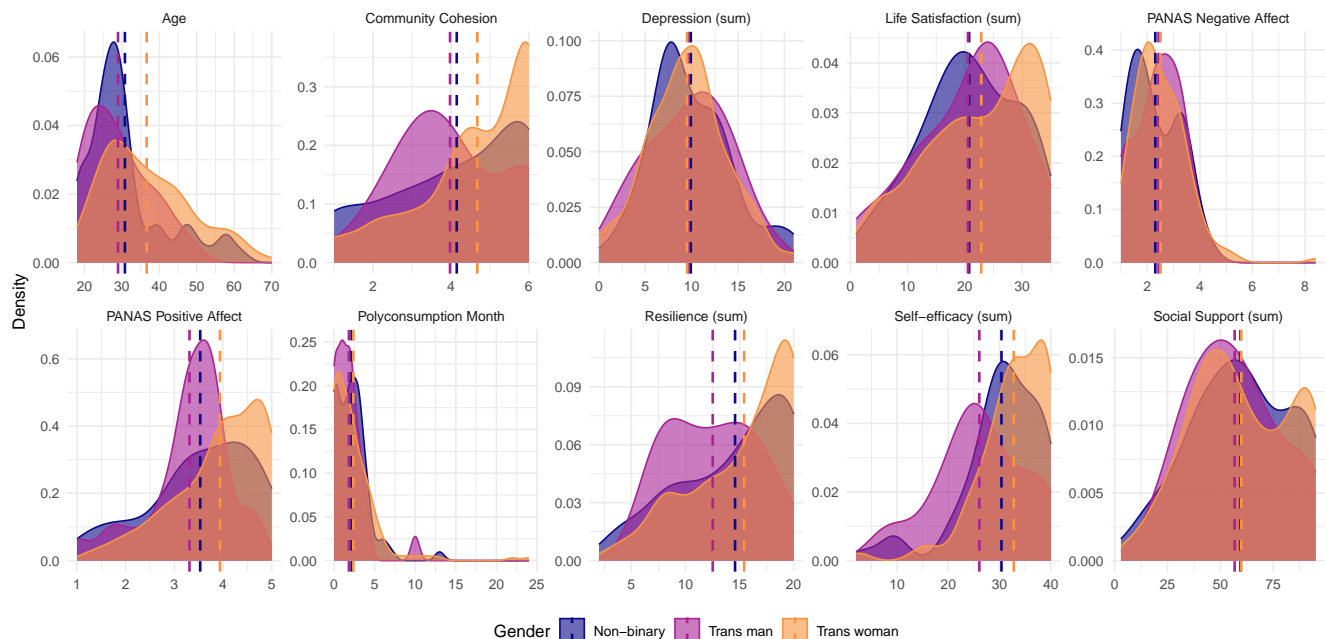

Figure S3. Kernel density distribution of measured numeric variables by gender. Coloured dashed lines represent mean values. For interpretability, Self-efficacy, Life Satisfaction, Resilience, Depression, and Social Support are shown as sum scores, calculated only for participants with complete item responses. Detailed descriptives are found in Table S3.

3.5   Table S4. Correlations

```
# Compute correlations for all participants combined
dat.corr.ALL <- data |>
```

```r
  select(Age, Polyconsumption_Month:Community_Cohesion, -Discrimination) |> # Numeric variables
  corr.stars() |>
  tail(-1) |>
  rownames_to_column(var = " ")

# Compute correlations for non-binary participants
dat.corr.NB <- data |>
  filter(Gender == "Non-binary") |> # Select only non binary
  select(Age, Polyconsumption_Month:Community_Cohesion, -Discrimination) |> # Numeric variables
  corr.stars() |> # Compute correlation matrix with significance stars
  tail(-1) |>
  rownames_to_column(var = " ") # Move row names to a column

# Compute correlations for trans men
dat.corr.TM <- data |>
  filter(Gender == "Trans man") |> # Select only trans men
  select(Age, Polyconsumption_Month:Community_Cohesion, -Discrimination) |> # Numeric variables
  corr.stars() |>
  tail(-1) |>
  rownames_to_column(var = " ")

# Compute correlations for trans women
dat.corr.TW <- data |>
  filter(Gender == "Trans woman") |> # Select only trans women
  select(Age, Polyconsumption_Month:Community_Cohesion, -Discrimination) |> # Numeric variables
  corr.stars() |>
  tail(-1) |>
  rownames_to_column(var = " ")

# Format and combine the correlation tables
bind_rows(dat.corr.ALL, dat.corr.NB, dat.corr.TM, dat.corr.TW) |>
  rename_with(~ gsub("Self_efficacy", "Self-efficacy", .x)) |>
  rename_with(~ gsub("_", " ", .x)) |>
  mutate_at(" ", str_replace_all, "Self_efficacy", "Self-efficacy") |>
  mutate_at(" ", str_replace_all, "_", " ") |>
  kable(
    digits = 2, booktabs = TRUE,
    align = c("l", rep("c", 12)),
    linesep = "",
    caption = "Correlations between measured variables", escape = FALSE
  ) |>
  # Add grouped row labels for each participant group
  pack_rows("All participants",
    start_row = 1, end_row = 12, bold = FALSE,
    background = "lightgray"
  ) |>
  pack_rows("Non binary",
    start_row = 13, end_row = 24, bold = FALSE,
    background = "lightgray"
  ) |>
  pack_rows("Trans men",
    start_row = 25, end_row = 36, bold = FALSE,
    background = "lightgray"
  ) |>
  pack_rows("Trans women",
    start_row = 37, end_row = 48, bold = FALSE,
```

```
    background = "lightgray"
) |>
# Apply LaTeX styling
kable_styling(latex_options = c("HOLD_position", "scale_down")) |>
column_spec(2:10, width = "2.2cm") |> # Adjust column widths
# Add footnote explaining correlation significance levels
footnote(
  general = paste0(
    "Values represent Spearman correlation coefficients ($\\\\rho$). ",
    "For significance, $^{\\\\dagger}p$ < 0.1, *$p$ < 0.05, ",
    "**$p$ < 0.01, ***$p$ < 0.001. ",
    "Significant correlations are in bold."
  ),
  threeparttable = TRUE, footnote_as_chunk = TRUE, escape = FALSE
) |>
landscape() # Rotate table for better readability in LaTeX
```

Table S4. *Correlations between measured variables*

	Age	Polyconsumption Month	Disease Burden	Group Membership	Community Engagement	Self-efficacy	Life Satisfaction	Resilience	Depression	Social Support	PANAS Positive Affect	PANAS Negative Affect
All participants												
Polyconsumption Month	**-0.32***											
Disease Burden	**0.16**	-0.03										
Group Membership	**0.17**	-0.04	0.11†									
Community Engagement	-0.01	0.01	0.07	**0.42***								
Self-efficacy	**0.23***	**-0.18**	-0.12†	**0.13***	0.06							
Life Satisfaction	0.11†	**-0.21***	**-0.23***	**0.22***	0.07	**0.4***						
Resilience	**0.17**	**-0.14***	-0.07	**0.29***	**0.14***	**0.49***	**0.49***					
Depression	0.05	0.10	**0.27***	**0.13***	0.10	-0.10	**-0.21***	-0.05				
Social Support	**0.14***	-0.06	-0.08	**0.24***	0.12†	**0.33***	**0.33***	**0.26***	-0.06			
PANAS Positive Affect	**0.18**	-0.10	-0.10	**0.26***	0.02	**0.45***	**0.46***	**0.42***	-0.02	**0.33***		
PANAS Negative Affect	-0.11†	**0.19**	0.11†	-0.06	**-0.12***	-0.09	**-0.18**	**-0.19**	**0.38***	0.04	-0.02	
Community Cohesion	**0.19**	-0.09	0.06	**0.26***	**0.25***	**0.28***	**0.25***	**0.29***	**0.23***	**0.26***	**0.3***	-0.03
Non binary												
Polyconsumption Month	**-0.28***											
Disease Burden	0.22†	0.01										
Group Membership	0.12	-0.02	0.06									
Community Engagement	-0.05	0.04	0.21†	**0.39***								
Self-efficacy	**0.35**	**-0.3***	-0.05	0.03	0.21							
Life Satisfaction	**0.33**	**-0.34**	-0.10	0.05	0.00	**0.26***						
Resilience	0.19	-0.21†	-0.04	0.17	0.19	**0.56***	**0.48***					
Depression	**0.31***	0.15	**0.47***	**0.32**	0.20	0.11	0.01	0.13				
Social Support	0.14	0.13	0.08	0.12	0.05	0.15	**0.32***	0.05	0.15			
PANAS Positive Affect	**0.27***	-0.06	-0.06	0.21†	0.10	**0.49***	**0.27***	**0.27***	0.20	0.22†		
PANAS Negative Affect	0.01	0.21†	0.10	-0.12	-0.02	-0.17	-0.13	**-0.26***	**0.35**	0.02	0.18	
Community Cohesion	**0.27***	0.02	0.17	**0.31***	**0.39**	**0.3***	0.14	0.06	**0.41**	0.23†	0.20	0.09
Trans men												
Polyconsumption Month	-0.24											
Disease Burden	-0.04	-0.19										
Group Membership	-0.07	-0.07	**0.47***									
Community Engagement	-0.15	0.18	0.22	**0.55**								
Self-efficacy	0.07	-0.24	0.02	**0.52***	0.05							
Life Satisfaction	0.19	0.21	-0.11	0.06	-0.19	0.21						
Resilience	-0.08	0.15	0.10	0.34†	-0.16	**0.57***	0.12					
Depression	-0.16	0.05	0.35	**0.49***	**0.62**	0.14	-0.21	-0.22				
Social Support	0.13	-0.37†	-0.05	0.4†	0.05	0.35	0.15	0.27	-0.14			
PANAS Positive Affect	0.27	-0.11	-0.10	0.31	-0.28	0.34	0.30	**0.61***	-0.29	0.44†		
PANAS Negative Affect	-0.27	0.16	0.27	0.17	0.13	-0.01	-0.29	-0.34†	**0.58**	-0.15	-0.32	
Community Cohesion	-0.15	-0.01	0.14	**0.4***	0.09	**0.5***	-0.06	0.34†	0.22	0.27	0.25	0.16
Trans women												
Polyconsumption Month	**-0.37***											
Disease Burden	**0.21**	-0.03										
Group Membership	**0.18***	-0.03	0.09									
Community Engagement	0.07	-0.02	-0.02	**0.42***								
Self-efficacy	0.08	-0.14†	-0.13†	0.13†	0.05							
Life Satisfaction	-0.03	**-0.22**	**-0.29***	**0.28***	**0.16***	**0.46***						
Resilience	0.12	**-0.16***	-0.09	**0.31***	**0.18***	**0.42***	**0.51***					
Depression	0.03	0.09	**0.19***	0.02	-0.01	-0.15†	**-0.3***	-0.07				
Social Support	0.14†	-0.10	-0.14†	**0.27***	0.14†	**0.43***	**0.35***	**0.32***	-0.11			
PANAS Positive Affect	0.06	-0.13†	-0.12	**0.24***	0.05	**0.41***	**0.5***	**0.41***	-0.05	**0.37***		
PANAS Negative Affect	**-0.18***	**0.19***	0.11	-0.09	**-0.2**	-0.06	**-0.21**	-0.13†	**0.37***	0.07	-0.10	
Community Cohesion	**0.16***	-0.13†	0.01	**0.2**	**0.23**	**0.21**	**0.27***	**0.33***	**0.18***	**0.27***	**0.3***	-0.10

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

4   Multi-model Inference

This section outlines the multi-model inference approach used to examine predictors of various outcome variables. The strategy follows a systematic model selection process, accounting for model uncertainty.

For each outcome variable, we repeated the following steps:

1. **Data Preparation**

   - Relevant predictors are selected from the cleaned dataset.

   - Missing values are removed to ensure complete case analysis[1]

2. **Global Model Specification**

   - A linear model (`lm`) is specified, including a broad set of demographic, psychological, and social predictors.

   - This full model represents all hypothesized influences on the outcome.

3. **Model Selection via `dredge()`**

   - The `MuMIn::dredge()` function generates all possible **nested models** from the global model.

   - Models are ranked by Akaike's Information Criterion corrected for small sample sizes ($AICc$).

   - This helps identify the most parsimonious models while considering model uncertainty.

4. **Model Averaging (`model.avg`)**

   - Models with $\Delta AICc < 2$ (within two $AICc$ units of the top model) are averaged.

   - This produces shrinkage estimates, ensuring robust coefficient estimates.

5. **Results Interpretation**

   - Tables summarize the coefficient estimates, importance values, and significance levels.

   - Figures visualize term estimates, confidence intervals, and predictor importance.

This process is repeated for each outcome variable in the following sections.

4.1   Life Satisfaction model

This section details the modelling process for Life Satisfaction, including data preparation, model selection using Akaike's Information Criterion corrected for small sample sizes ($AICc$), and model averaging based on the best-ranked models.

The global model includes a broad set of predictors representing demographic, psychological, and social factors, allowing us to explore their relative contributions to Life Satisfaction. The model selection procedure identifies the most parsimonious models while accounting for model uncertainty.

```
dat_LS <- data |>
  select(
    Life_Satisfaction, Age, Gender, Ethnicity, Marital_Status, Education, Housing,
    Self_efficacy, Community_Cohesion, Depression, Social_Support, Polyconsumption_Month,
    Disease_Burden, Discrimination, Group_Membership, Community_Engagement
  ) |>
  drop_na()

global_LS <- lm(
  Life_Satisfaction ~ Age + Gender + Ethnicity + Marital_Status + Education +
    Housing + Self_efficacy + Community_Cohesion + Depression +
    Social_Support + Polyconsumption_Month + Disease_Burden + Discrimination +
    Group_Membership + Community_Engagement,
```

---

[1]To maximize the number of non-missing cases, we created separate datasets for each outcome variable. Since models generated by `dredge()` must not contain missing values (NA) to remain comparable, this approach ensures that each model includes the largest possible sample size.

```
  data = dat_LS,
  na.action = "na.fail"
)
```

4.1.1   Dredge

To identify the best-fitting models, we perform automated model selection using `MuMIn::dredge()`, which fits all possible models nested within the global model and ranks them based on *AICc*. This helps determine the most important predictors of Life Satisfaction while considering model uncertainty.

```
dr_LS <- dredge(
  global_LS,
  # trace = 2 # uncomment to see progress bar
)
```

4.1.1.1   Fig. S4.   Dredge results of the Life Satisfaction model   The plot below visualizes the model selection results, displaying the top 100 models ranked by *AICc*. Each row represents an individual model, with blue cells indicating included predictors. The height of each row reflects the model's Akaike weight (ω), representing its relative support in the model set.

```
LS_dredge_plot <- dredge.plot(dr_LS) # Run the function and store the result
```
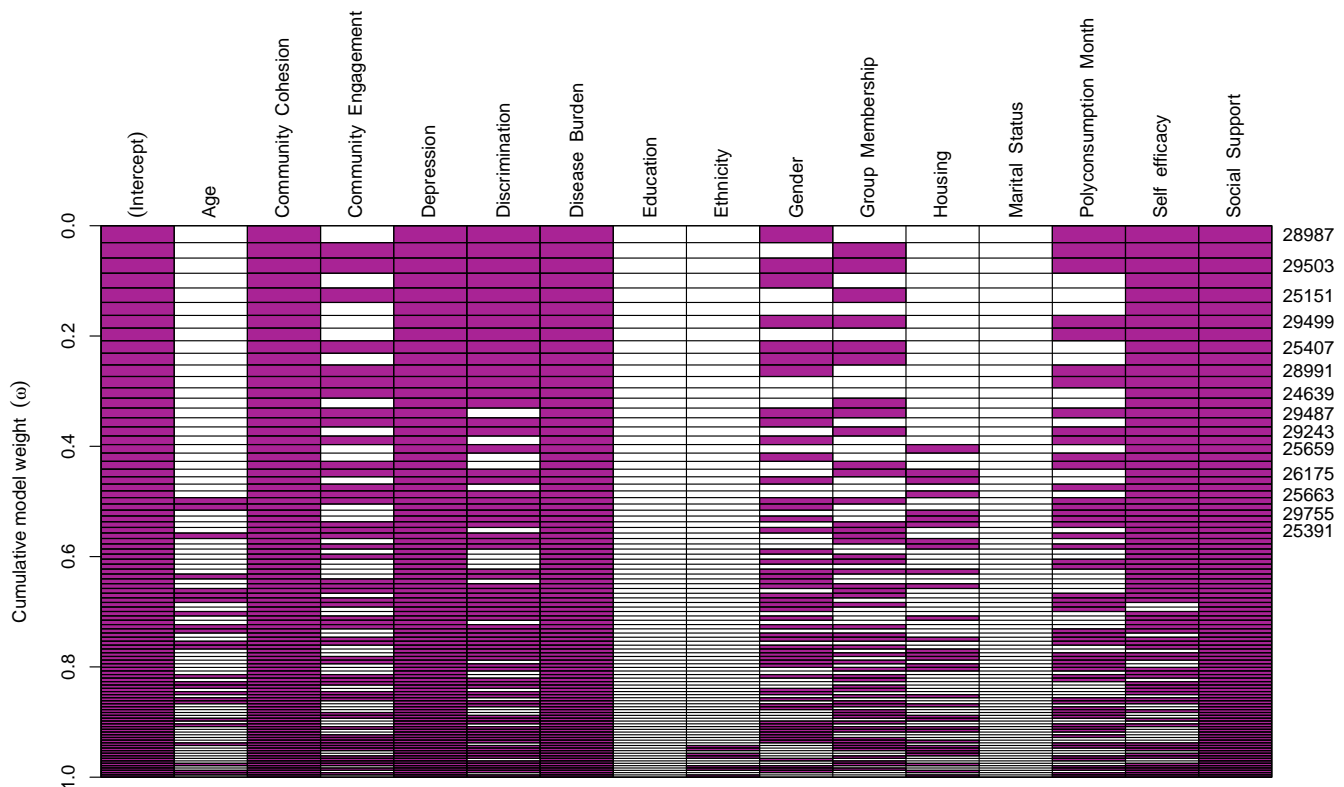

Figure S4. Model selection plot. Each row represents one of the top 100 models (ranked by *AICc*) out of 32,768 fitted models. Purple cells indicate included terms, with row height corresponding to each model's relative support based on its Akaike weight (
omega)

4.1.2　Average model

To account for model uncertainty, we compute an averaged model that combines estimates from all models with $\Delta AICc < 2$ (i.e., models within two $AICc$ units of the best-supported model). This approach produces more robust and conservative parameter estimates, reducing overconfidence in any single model.

```
avg_LS <- model.avg(dr_LS, subset = delta < 2, fit = TRUE)
```

4.1.2.1　Table S5. Term estimates for the Life Satisfaction model　The table below presents coefficient estimates from the full average model, including 95% confidence intervals, standard errors, and significance levels.

```
tab_avg_LS <- avg.mod.summary(avg_LS, data = dat_LS)
tab_avg_LS$table
```
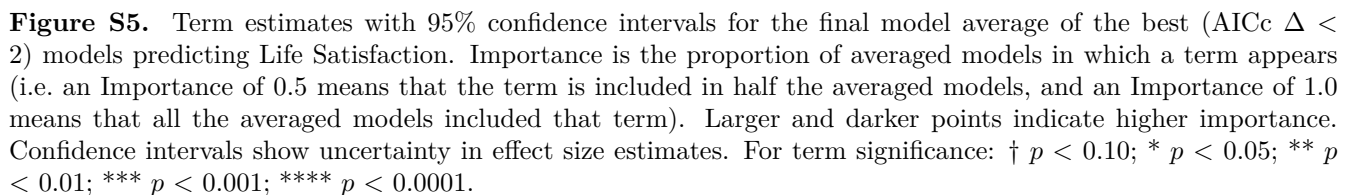
Table S5. *Coefficient estimates for the Life Satisfaction model.*

Term	B	$95\%CIs$	z	p
(Intercept)	2.935	1.394 — 4.476	3.733	**< 0.001**
Age	0.000	-0.004 — 0.003	0.090	0.93
Community Cohesion	0.237	0.086 — 0.389	3.073	**0.0021**
Community Engagement	-0.061	-0.212 — 0.091	0.784	0.43
Depression	-0.572	-0.943 — -0.202	3.028	**0.0025**
Discrimination: Yes	-0.541	-1.276 — 0.195	1.441	0.15
Disease Burden	-0.234	-0.397 — -0.071	2.815	**0.0049**
Gender: Trans man	0.281	-0.504 — 1.067	0.702	0.48
Gender: Trans woman	0.247	-0.327 — 0.82	0.843	0.4
Group Membership	0.075	-0.136 — 0.285	0.696	0.49
Housing: Living with family	-0.144	-1.033 — 0.746	0.316	0.75
Housing: Renting (entire home)	-0.043	-0.419 — 0.332	0.227	0.82
Housing: Shared rental (room)	-0.066	-0.538 — 0.405	0.275	0.78
Housing: Without permanent housing	-0.107	-0.827 — 0.613	0.291	0.77
Polyconsumption Month	-0.031	-0.105 — 0.044	0.800	0.42
Self-efficacy	0.413	0.043 — 0.784	2.189	**0.0286**
Social Support	0.313	0.121 — 0.505	3.190	**0.0014**

Note:　This table includes estimates based on a full model average. The full model average ensures that all variables are included in every model, with coefficients set to zero when absent. As a consequence, it acts as a shrinkage estimator, making estimates more conservative. B represents unstandardized model coefficients. Reference levels: Gender = Non-binary; Housing = Home-owner; Discrimination = No. Significant effects are in bold.

4.1.2.2　Fig. S5. Term estimates for the Life Satisfaction model　This figure represents parameter estimates from the model-averaged coefficients.

```
fig_avg_LS <- avg.mod.plot(avg_LS)
fig_avg_LS
```

Figure S5. Term estimates with 95% confidence intervals for the final model average of the best (AICc Δ < 2) models predicting Life Satisfaction. Importance is the proportion of averaged models in which a term appears (i.e. an Importance of 0.5 means that the term is included in half the averaged models, and an Importance of 1.0 means that all the averaged models included that term). Larger and darker points indicate higher importance. Confidence intervals show uncertainty in effect size estimates. For term significance: † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 0.0001$.

4.2   PANAS Positive Affect model

This section details the modelling process for PANAS Positive Affect, including data preparation, model selection using Akaike's Information Criterion corrected for small sample sizes ($AICc$), and model averaging based on the best-ranked models.

The global model includes a broad set of predictors representing demographic, psychological, and social factors, allowing us to explore their relative contributions to PANAS Positive Affect. The model selection procedure identifies the most parsimonious models while accounting for model uncertainty.

```
dat_PANAS_P <- data |>
  select(
    PANAS_Positive_Affect, Age, Gender, Ethnicity, Marital_Status, Education, Housing,
    Self_efficacy, Community_Cohesion, Depression, Social_Support, Polyconsumption_Month,
    Disease_Burden, Discrimination, Group_Membership, Community_Engagement
  ) |>
  drop_na()
```

```r
global_PANAS_P <- lm(
  PANAS_Positive_Affect ~ Age + Gender + Ethnicity + Marital_Status + Education +
    Housing + Self_efficacy + Community_Cohesion + Depression +
    Social_Support + Polyconsumption_Month + Disease_Burden + Discrimination +
    Group_Membership + Community_Engagement,
  data = dat_PANAS_P,
  na.action = "na.fail"
)
```

4.2.1   Dredge

To identify the best-fitting models, we perform automated model selection using `MuMIn::dredge()`, which fits all possible models nested within the global model and ranks them based on *AICc*. This helps determine the most important predictors of PANAS Positive Affect while considering model uncertainty.

```r
dr_PANAS_P <- dredge(
  global_PANAS_P,
  # trace = 2 # uncomment to see progress bar
)
```

4.2.1.1   Fig. S6. Dredge results of the PANAS Positive Affect model

The plot below visualizes the model selection results, displaying the top 100 models ranked by *AICc*. Each row represents an individual model, with blue cells indicating included predictors. The height of each row reflects the model's Akaike weight (ω), representing its relative support in the model set.

```r
PANAS_P_dredge_plot <- dredge.plot(dr_PANAS_P) # Run the function and store the result
```
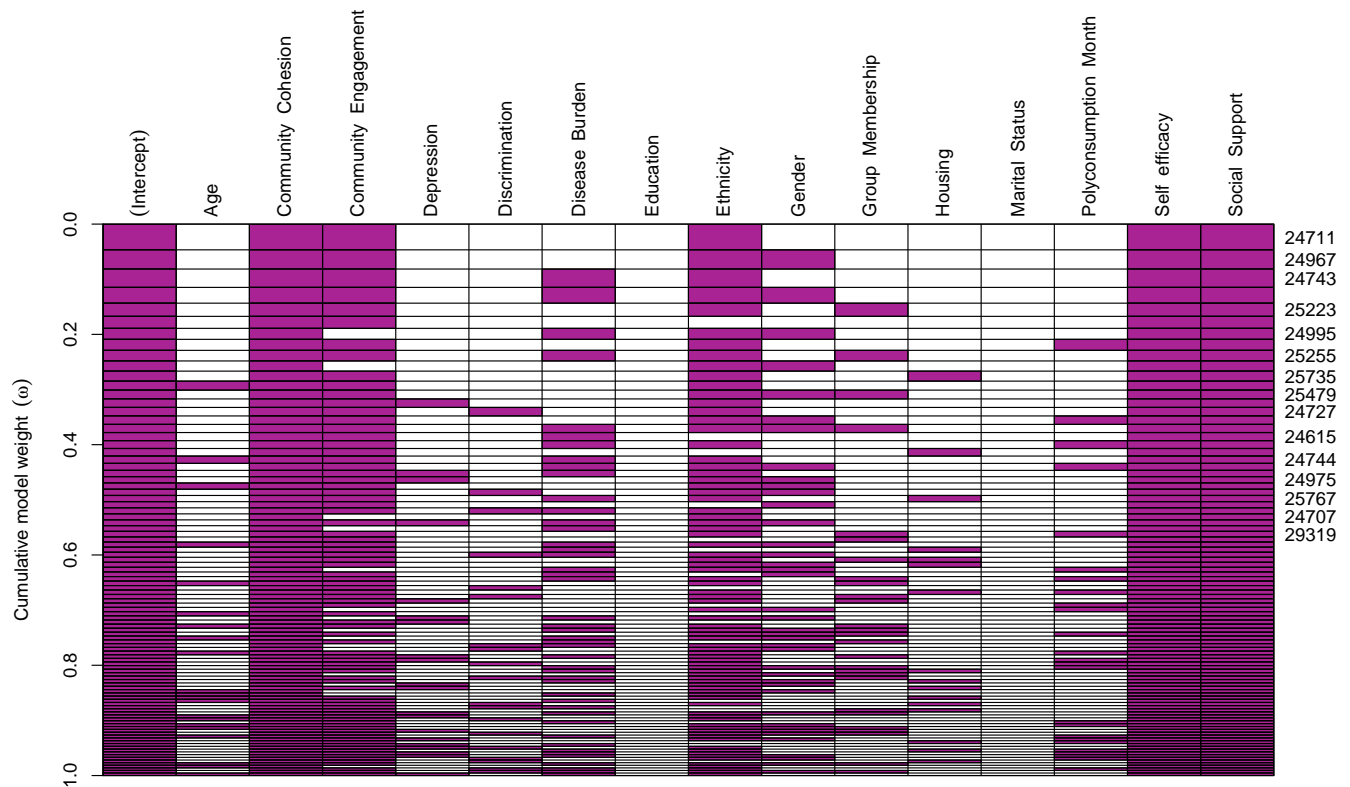
Figure S6. Model selection plot. Each row represents one of the top 100 models (ranked by *AICc*) out of 32,768 fitted models. Purple cells indicate included terms, with row height corresponding to each model's relative support based on its Akaike weight (
omega)

4.2.2 Average model

To account for model uncertainty, we compute an averaged model that combines estimates from all models with $\Delta AICc < 2$ (i.e., models within two *AICc* units of the best-supported model). This approach produces more robust and conservative parameter estimates, reducing overconfidence in any single model.

```
avg_PANAS_P <- model.avg(dr_PANAS_P, subset = delta < 2, fit = TRUE)
```

4.2.2.1 Table S6. Term estimates for the PANAS Positive Affect model   The table below presents coefficient estimates from the full average model, including 95% confidence intervals, standard errors, and significance levels.

```
tab_avg_PANAS_P <- avg.mod.summary(avg_PANAS_P, data = dat_PANAS_P)
tab_avg_PANAS_P$table
```

Table S6. *Coefficient estimates for the PANAS Positive Affect model.*

Term	B	$95\%CIs$	z	p
(Intercept)	0.649	-0.118 — 1.416	1.658	0.1
Community Cohesion	0.127	0.044 — 0.21	2.998	**0.0027**
Community Engagement	-0.073	-0.167 — 0.02	1.535	0.12
Disease Burden	-0.021	-0.099 — 0.056	0.538	0.59
Ethnicity: Indigenous	0.311	-0.134 — 0.756	1.369	0.17
Ethnicity: None	0.248	-0.051 — 0.547	1.625	0.1
Ethnicity: Rrom	0.817	0.014 — 1.62	1.995	**0.046**
Gender: Trans man	0.028	-0.252 — 0.309	0.198	0.84
Gender: Trans woman	0.098	-0.204 — 0.399	0.634	0.53
Group Membership	0.009	-0.053 — 0.07	0.281	0.78
Housing: Living with family	0.004	-0.156 — 0.164	0.051	0.96
Housing: Renting (entire home)	0.027	-0.208 — 0.262	0.224	0.82
Housing: Shared rental (room)	0.011	-0.136 — 0.158	0.144	0.89
Housing: Without permanent housing	0.024	-0.225 — 0.272	0.188	0.85
Polyconsumption Month	-0.001	-0.013 — 0.011	0.153	0.88
Self-efficacy	0.584	0.379 — 0.789	5.589	**< 0.0001**
Social Support	0.185	0.077 — 0.294	3.356	**< 0.001**

Note: This table includes estimates based on a full model average. The full model average ensures that all variables are included in every model, with coefficients set to zero when absent. As a consequence, it acts as a shrinkage estimator, making estimates more conservative. B represents unstandardized model coefficients. Reference levels: Gender = Non-binary; Ethnicity = Afro-Colombian; Housing = Home-owner. Significant effects are in bold.

4.2.2.2   Fig. S7. Term estimates for the PANAS Positive Affect model   This figure represents parameter estimates from the model-averaged coefficients.

```
fig_avg_PANAS_P <- avg.mod.plot(avg_PANAS_P)
fig_avg_PANAS_P
```
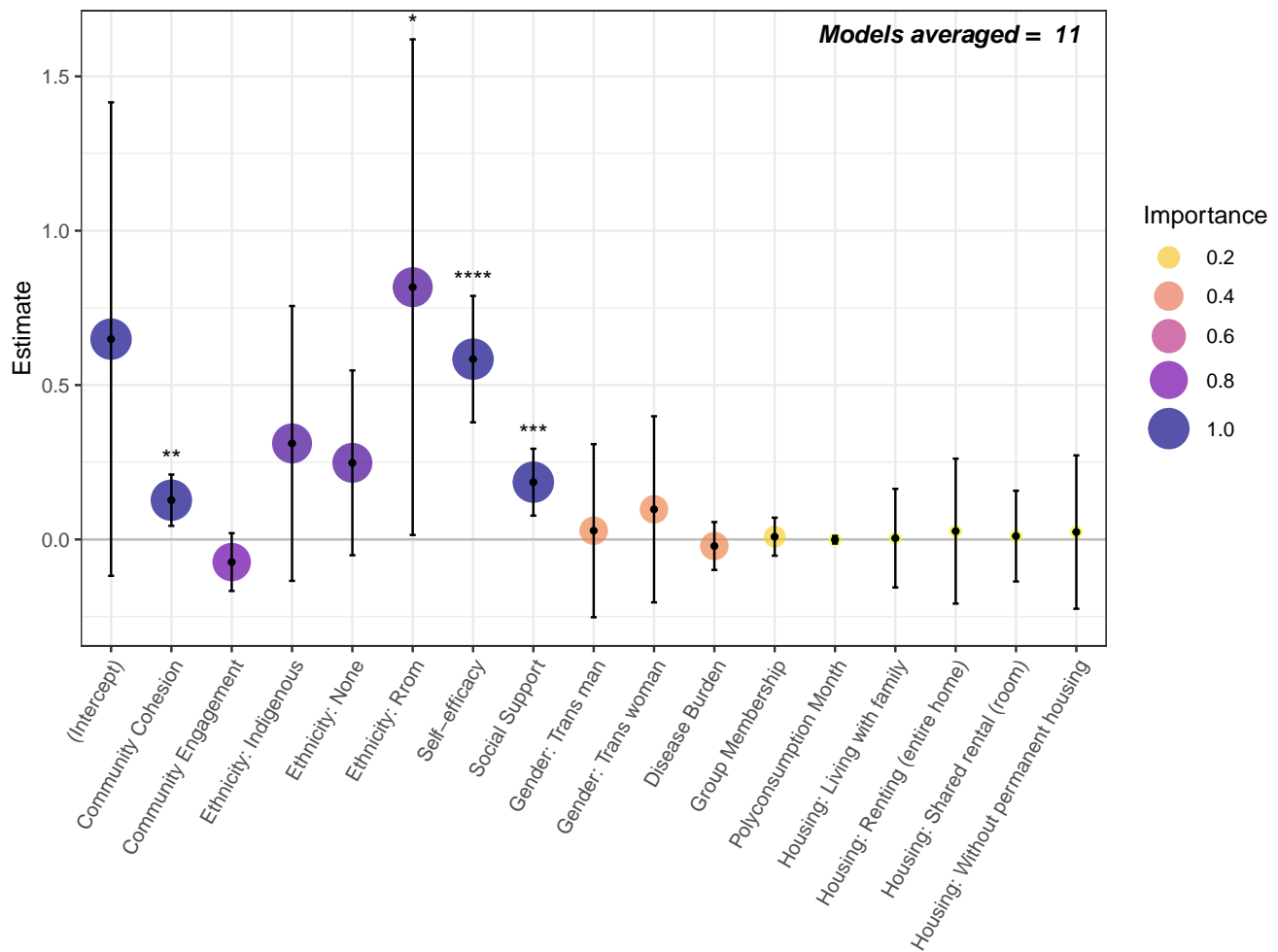
Figure S7. Term estimates with 95% confidence intervals for the final model average of the best (AICc Δ < 2) models predicting PANAS Positive Affect. Importance is the proportion of averaged models in which a term appears (i.e. an Importance of 0.5 means that the term is included in half the averaged models, and an Importance of 1.0 means that all the averaged models included that term). Larger and darker points indicate higher importance. Confidence intervals show uncertainty in effect size estimates. For term significance: † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 0.0001$.

4.3   PANAS Negative Affect model

This section details the modelling process for PANAS Negative Affect, including data preparation, model selection using Akaike's Information Criterion corrected for small sample sizes ($AICc$), and model averaging based on the best-ranked models.

The global model includes a broad set of predictors representing demographic, psychological, and social factors, allowing us to explore their relative contributions to PANAS Negative Affect. The model selection procedure identifies the most parsimonious models while accounting for model uncertainty.

```
dat_PANAS_N <- data |>
  select(
    PANAS_Negative_Affect, Age, Gender, Ethnicity, Marital_Status, Education, Housing,
    Self_efficacy, Community_Cohesion, Depression, Social_Support, Polyconsumption_Month,
    Disease_Burden, Discrimination, Group_Membership, Community_Engagement
  ) |>
  drop_na()
```

```
global_PANAS_N <- lm(
  PANAS_Negative_Affect ~ Age + Gender + Ethnicity + Marital_Status + Education +
    Housing + Self_efficacy + Community_Cohesion + Depression +
    Social_Support + Polyconsumption_Month + Disease_Burden + Discrimination +
    Group_Membership + Community_Engagement,
  data = dat_PANAS_N,
  na.action = "na.fail"
)
```

4.3.1   Dredge

To identify the best-fitting models, we perform automated model selection using `MuMIn::dredge()`, which fits all possible models nested within the global model and ranks them based on *AICc*. This helps determine the most important predictors of PANAS Negative Affect while considering model uncertainty.

```
dr_PANAS_N <- dredge(
  global_PANAS_N,
  # trace = 2 # uncomment to see progress bar
)
```

4.3.1.1   Fig. S8. Dredge results of the PANAS Negative Affect model   The plot below visualizes the model selection results, displaying the top 100 models ranked by *AICc*. Each row represents an individual model, with blue cells indicating included predictors. The height of each row reflects the model's Akaike weight (ω), representing its relative support in the model set.

```
PANAS_N_dredge_plot <- dredge.plot(dr_PANAS_N) # Run the function and store the result
```
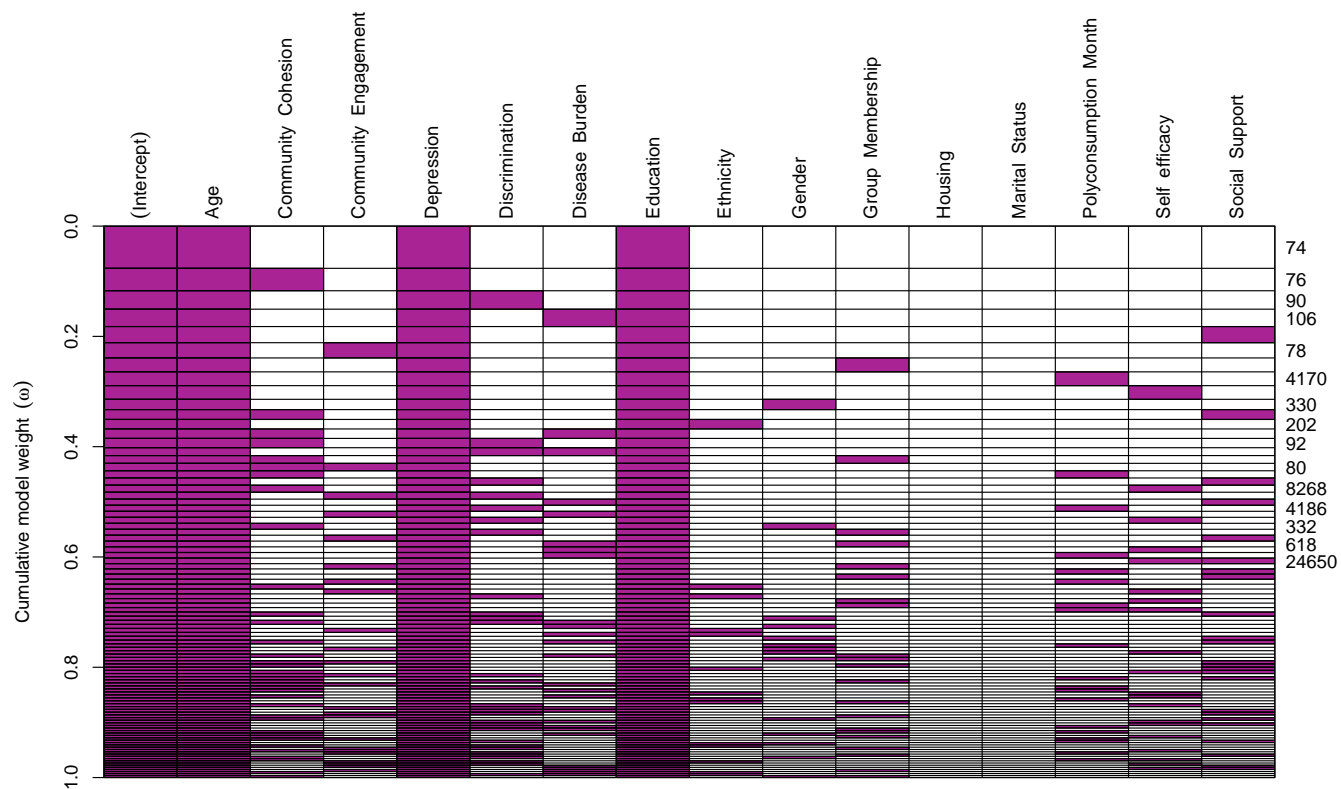
Figure S8. Model selection plot. Each row represents one of the top 100 models (ranked by *AICc*) out of 32,768 fitted models. Purple cells indicate included terms, with row height corresponding to each model's relative support based on its Akaike weight (
omega)

4.3.2 Average model

To account for model uncertainty, we compute an averaged model that combines estimates from all models with $\Delta AICc < 2$ (i.e., models within two *AICc* units of the best-supported model). This approach produces more robust and conservative parameter estimates, reducing overconfidence in any single model.

```
avg_PANAS_N <- model.avg(dr_PANAS_N, subset = delta < 2, fit = TRUE)
```

4.3.2.1 Table S7. Term estimates for the PANAS Negative Affect model   The table below presents coefficient estimates from the full average model, including 95% confidence intervals, standard errors, and significance levels.

```
tab_avg_PANAS_N <- avg.mod.summary(avg_PANAS_N, data = dat_PANAS_N)
tab_avg_PANAS_N$table
```

Table S7. *Coefficient estimates for the PANAS Negative Affect model.*

Term	B	95%CIs	z	p
(Intercept)	2.768	1.605 — 3.932	4.663	**< 0.0001**
Age	-0.019	-0.03 — -0.008	3.426	**< 0.001**
Community Cohesion	-0.009	-0.063 — 0.045	0.321	0.75
Depression	0.692	0.481 — 0.904	6.415	**< 0.0001**
Discrimination: Yes	0.024	-0.163 — 0.211	0.249	0.8
Disease Burden	-0.005	-0.053 — 0.042	0.225	0.82
Education: No studies, literate	0.247	-1.306 — 1.8	0.312	0.76
Education: Primary school	-0.591	-1.679 — 0.496	1.066	0.29
Education: Primary school (unfinished)	-0.177	-1.239 — 0.885	0.326	0.74
Education: Secondary school	-0.638	-1.649 — 0.372	1.238	0.22
Education: Secondary school (unfinished)	-0.679	-1.696 — 0.337	1.310	0.19
Education: Technical degree	-1.327	-2.378 — -0.275	2.472	**0.0134**
Education: University	-1.619	-2.87 — -0.368	2.537	**0.0112**
Education: University (unfinished)	-1.081	-2.169 — 0.007	1.948	0.05
Social Support	0.004	-0.041 — 0.05	0.190	0.85

Note:　　This table includes estimates based on a full model average. The full model average ensures that all variables are included in every model, with coefficients set to zero when absent. As a consequence, it acts as a shrinkage estimator, making estimates more conservative. B represents unstandardized model coefficients. Reference levels: Education = No studies, illiterate; Discrimination = No. Significant effects are in bold.

4.3.2.2　Fig. S9. Term estimates for the PANAS Negative Affect model　This figure represents parameter estimates from the model-averaged coefficients.

```
fig_avg_PANAS_N <- avg.mod.plot(avg_PANAS_N)
fig_avg_PANAS_N
```
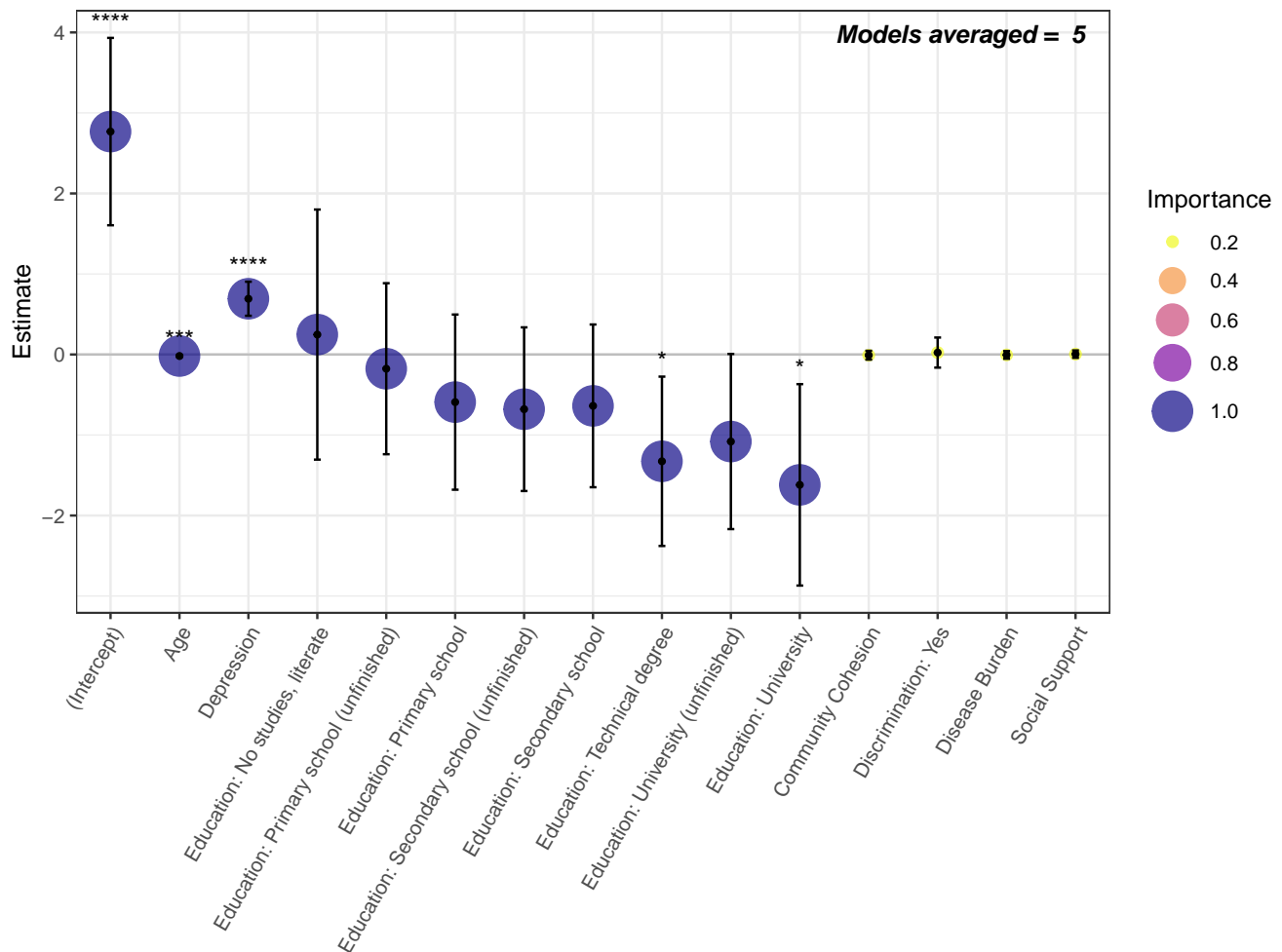
Figure S9. Term estimates with 95% confidence intervals for the final model average of the best (AICc Δ < 2) models predicting PANAS Negative Affect. Importance is the proportion of averaged models in which a term appears (i.e. an Importance of 0.5 means that the term is included in half the averaged models, and an Importance of 1.0 means that all the averaged models included that term). Larger and darker points indicate higher importance. Confidence intervals show uncertainty in effect size estimates. For term significance: † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 0.0001$.

5   Final figures and tables

5.1   Fig. 1. Distribution of participants' measured variables by gender

```
ggarrange(p_desc1 +
          facet_wrap(~Variable, ncol = 3, scales = "free"),
        p_desc2 +
          facet_wrap(~Variable, ncol = 2, scales = "free",
                    labeller = label_wrap_gen(width = 17)),
        common.legend = TRUE,
        legend = "bottom",
        labels = "auto",
        ncol = 2,
        widths = c(2,1))
```
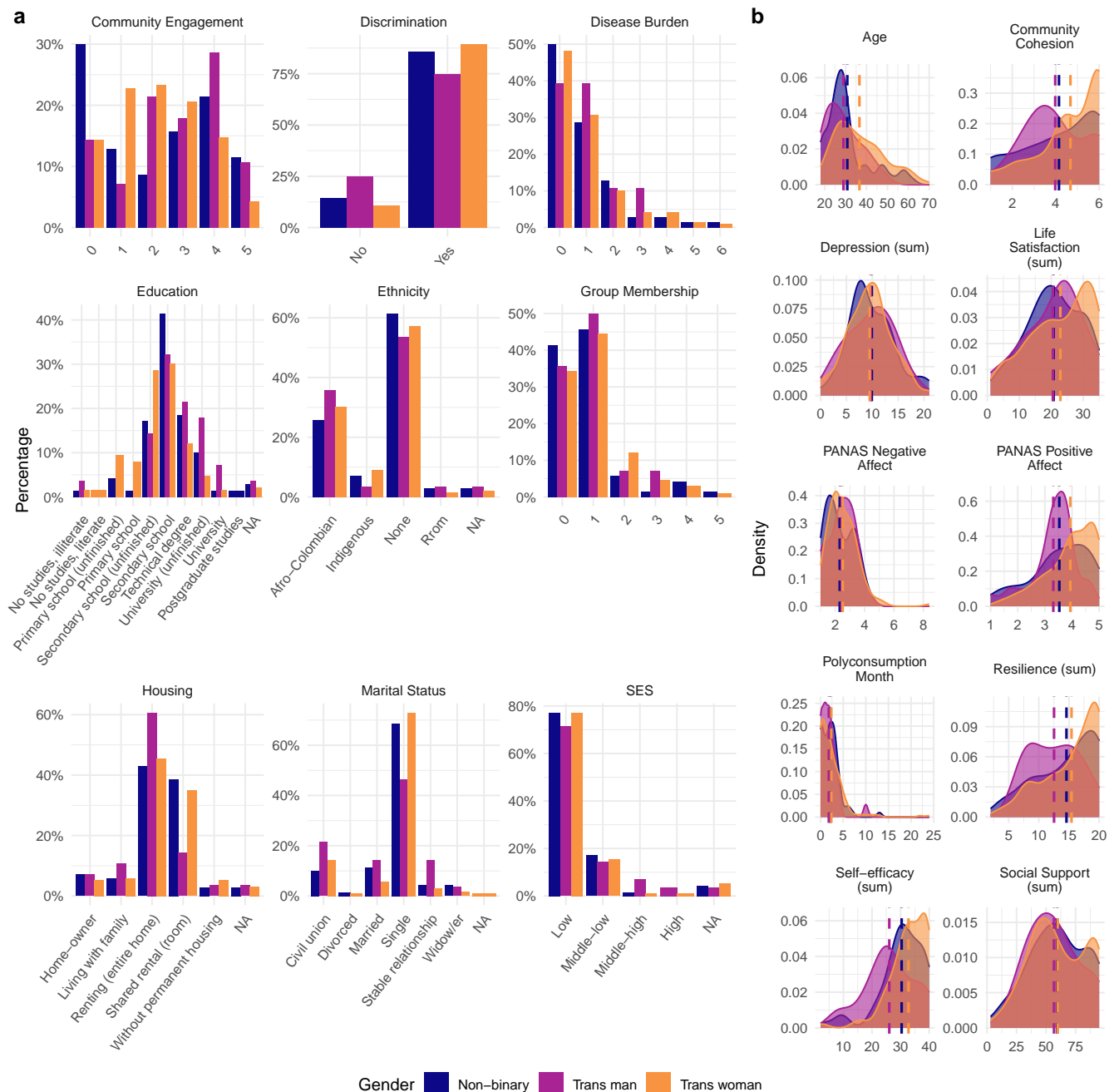
Figure 1. Distribution of variables by gender. **(a)** Categorical and ordinal variables. Values represent the percentage of responses in each category. To facilitate comparison across genders, percentages were calculated within each gender group. **(b)** Kernel density distribution of measured numeric variables. Coloured dashed lines represent mean values. For interpretability, Self-efficacy, Life Satisfaction, Resilience, Depression, and Social Support are shown as sum scores, calculated only for participants with complete item responses. Additional descriptive statistics are provided in Tables S2 and S3.

5.2   Fig. 2. Distribution of participants' measured variables by gender

```
ggarrange(as.ggplot(image_read_pdf("PDF_plots/Life_Satisfaction.pdf")),
          as.ggplot(image_read_pdf("PDF_plots/PANAS_Positive_Affect.pdf")),
          as.ggplot(image_read_pdf("PDF_plots/PANAS_Negative_Affect.pdf")),
          labels = "auto",
```
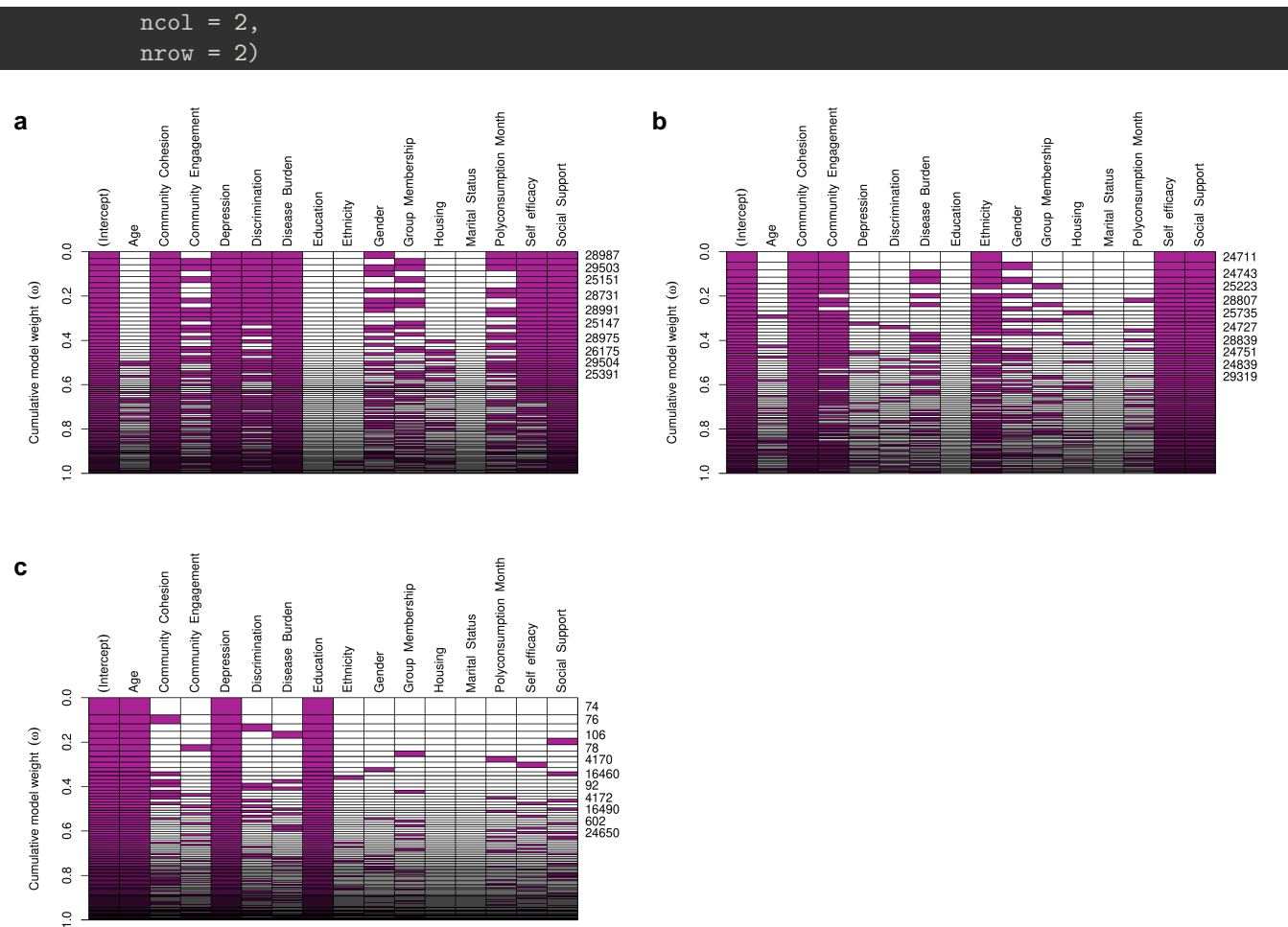
```
        ncol = 2,
        nrow = 2)
```


Figure 2. Model selection plot. **(a)** Life Satisfaction; **(b)** PANAS Positive Affect; **(c)** PANAS Negative Affect. In all cases, each row represents one of the top 100 candidate models (ranked by *AICc*) out of 32,768 fitted models. Purple cells indicate included terms, with row height corresponding to each model's relative support based on its Akaike weight (ω). Numbers on the right identify the top models.

```
bla <- ggplot() +
  background_image(image_read_pdf("PDF_plots/Life_Satisfaction.pdf", density = 600)) +
  theme(plot.margin = margin(t=0.2, l=1, r=0, b=0, unit = "cm"))
```

5.3   Tab. 1. Term estimates for the three final averaged model

```
tab_avg_LS$coefficients |>
  full_join(tab_avg_PANAS_P$coefficients, by = "term")  |>
  full_join(tab_avg_PANAS_N$coefficients, by = "term") |>
  arrange(term) |>
  kable(digits = 3,
        booktabs = TRUE,
        linesep = "",
        align = c("l", rep("c", 12)),
        caption = "Coefficient estimates for the three final average models",
        col.names = c("Term",
                      rep(c("$B$", "$95 \\% CIs$", "$z$", "$p$"), times = 3)),
```

```
        escape = FALSE
) |>
kable_styling(latex_options = c("scale_down", "HOLD_position")) |>
add_header_above(c(" ",
                    "Life Satisfaction" = 4,
                    "PANAS Positive Affect" = 4,
                    "PANAS Negative Affect" = 4)) |>
column_spec(c(5, 9), border_right = TRUE) |>
footnote(
  general = paste0(
    "This table includes estimates based on full model averages. ",
    "The full model average ensures that all predictors present in any
    averaged model are included, with coefficients set to zero when absent.
    As a consequence, it acts as a shrinkage ",
    "estimator, making estimates more conservative. ",
    "$B$ represents unstandardized model coefficients for each of the three models:
    Life Satisfaction (", tab_avg_LS$reference_levels, "),
    PANAS Positive Afffect(", tab_avg_PANAS_P$reference_levels, "),
    and PANAS Negative Afffect(", tab_avg_PANAS_N$reference_levels, ").",
    " Significant effects are in bold."
  ),
  threeparttable = TRUE,
  footnote_as_chunk = TRUE,
  escape = FALSE
)
```

Table 1. *Coefficient estimates for the three final average models*

	Life Satisfaction				PANAS Positive Affect				PANAS Negative Affect			
Term	B	95%CIs	z	p	B	95%CIs	z	p	B	95%CIs	z	p
(Intercept)	2.935	1.394 — 4.476	3.733	**< 0.001**	0.649	-0.118 — 1.416	1.658	0.1	2.768	1.605 — 3.932	4.663	**< 0.0001**
Age	0.000	-0.004 — 0.003	0.090	0.93					-0.019	-0.03 — -0.008	3.426	**< 0.001**
Community Cohesion	0.237	0.086 — 0.389	3.073	**0.0021**	0.127	0.044 — 0.21	2.998	**0.0027**	-0.009	-0.063 — 0.045	0.321	0.75
Community Engagement	-0.061	-0.212 — 0.091	0.784	0.43	-0.073	-0.167 — 0.02	1.535	0.12				
Depression	-0.572	-0.943 — -0.202	3.028	**0.0025**					0.692	0.481 — 0.904	6.415	**< 0.0001**
Discrimination: Yes	-0.541	-1.276 — 0.195	1.441	0.15					0.024	-0.163 — 0.211	0.249	0.8
Disease Burden	-0.234	-0.397 — -0.071	2.815	**0.0049**	-0.021	-0.099 — 0.056	0.538	0.59	-0.005	-0.053 — 0.042	0.225	0.82
Education: No studies, literate									0.247	-1.306 — 1.8	0.312	0.76
Education: Primary school									-0.591	-1.679 — 0.496	1.066	0.29
Education: Primary school (unfinished)									-0.177	-1.239 — 0.885	0.326	0.74
Education: Secondary school									-0.638	-1.649 — 0.372	1.238	0.22
Education: Secondary school (unfinished)									-0.679	-1.696 — 0.337	1.310	0.19
Education: Technical degree									-1.327	-2.378 — -0.275	2.472	**0.0134**
Education: University									-1.619	-2.87 — -0.368	2.537	**0.0112**
Education: University (unfinished)									-1.081	-2.169 — 0.007	1.948	0.05
Ethnicity: Indigenous					0.311	-0.134 — 0.756	1.369	0.17				
Ethnicity: None					0.248	-0.051 — 0.547	1.625	0.1				
Ethnicity: Rrom					0.817	0.014 — 1.62	1.995	**0.046**				
Gender: Trans man	0.281	-0.504 — 1.067	0.702	0.48	0.028	-0.252 — 0.309	0.198	0.84				
Gender: Trans woman	0.247	-0.327 — 0.82	0.843	0.4	0.098	-0.204 — 0.399	0.634	0.53				
Group Membership	0.075	-0.136 — 0.285	0.696	0.49	0.009	-0.053 — 0.07	0.281	0.78				
Housing: Living with family	-0.144	-1.033 — 0.746	0.316	0.75	0.004	-0.156 — 0.164	0.051	0.96				
Housing: Renting (entire home)	-0.043	-0.419 — 0.332	0.227	0.82	0.027	-0.208 — 0.262	0.224	0.82				
Housing: Shared rental (room)	-0.066	-0.538 — 0.405	0.275	0.78	0.011	-0.136 — 0.158	0.144	0.89				
Housing: Without permanent housing	-0.107	-0.827 — 0.613	0.291	0.77	0.024	-0.225 — 0.272	0.188	0.85				
Polyconsumption Month	-0.031	-0.105 — 0.044	0.800	0.42	-0.001	-0.013 — 0.011	0.153	0.88				
Self-efficacy	0.413	0.043 — 0.784	2.189	**0.0286**	0.584	0.379 — 0.789	5.589	**< 0.0001**				
Social Support	0.313	0.121 — 0.505	3.190	**0.0014**	0.185	0.077 — 0.294	3.356	**< 0.001**	0.004	-0.041 — 0.05	0.190	0.85

Note: This table includes estimates based on full model averages. The full model average ensures that all predictors present in any averaged model are included, with coefficients set to zero when absent. As a consequence, it acts as a shrinkage estimator, making estimates more conservative. *B* represents unstandardized model coefficients for each of the three models: Life Satisfaction (Reference levels: Gender = Non-binary; Housing = Home-owner; Discrimination = No), PANAS Positive Afffect(Reference levels: Gender = Non-binary; Ethnicity = Afro-Colombian; Housing = Home-owner), and PANAS Negative Afffect(Reference levels: Education = No studies, illiterate; Discrimination = No). Significant effects are in bold.

5.4   Fig. **3**. Term estimates for the three final averaged model

This figure represents parameter estimates from the model-averaged coefficients.

```
ggarrange(
  fig_avg_LS +
```

```r
    labs(subtitle = "Life Satistaction") +
    theme(legend.position = "none"),
  fig_avg_PANAS_P +
    labs(subtitle = "PANAS Possitive Affect") +
    theme(legend.position = "none"),
  fig_avg_PANAS_N +
    labs(subtitle = "PANAS Negative Affect") +
    theme(legend.position = "none"),
  as_ggplot(get_legend(fig_avg_LS)),
  ncol = 2,
  nrow = 2,
  labels = c("a", "b", "c", NULL)
)
```
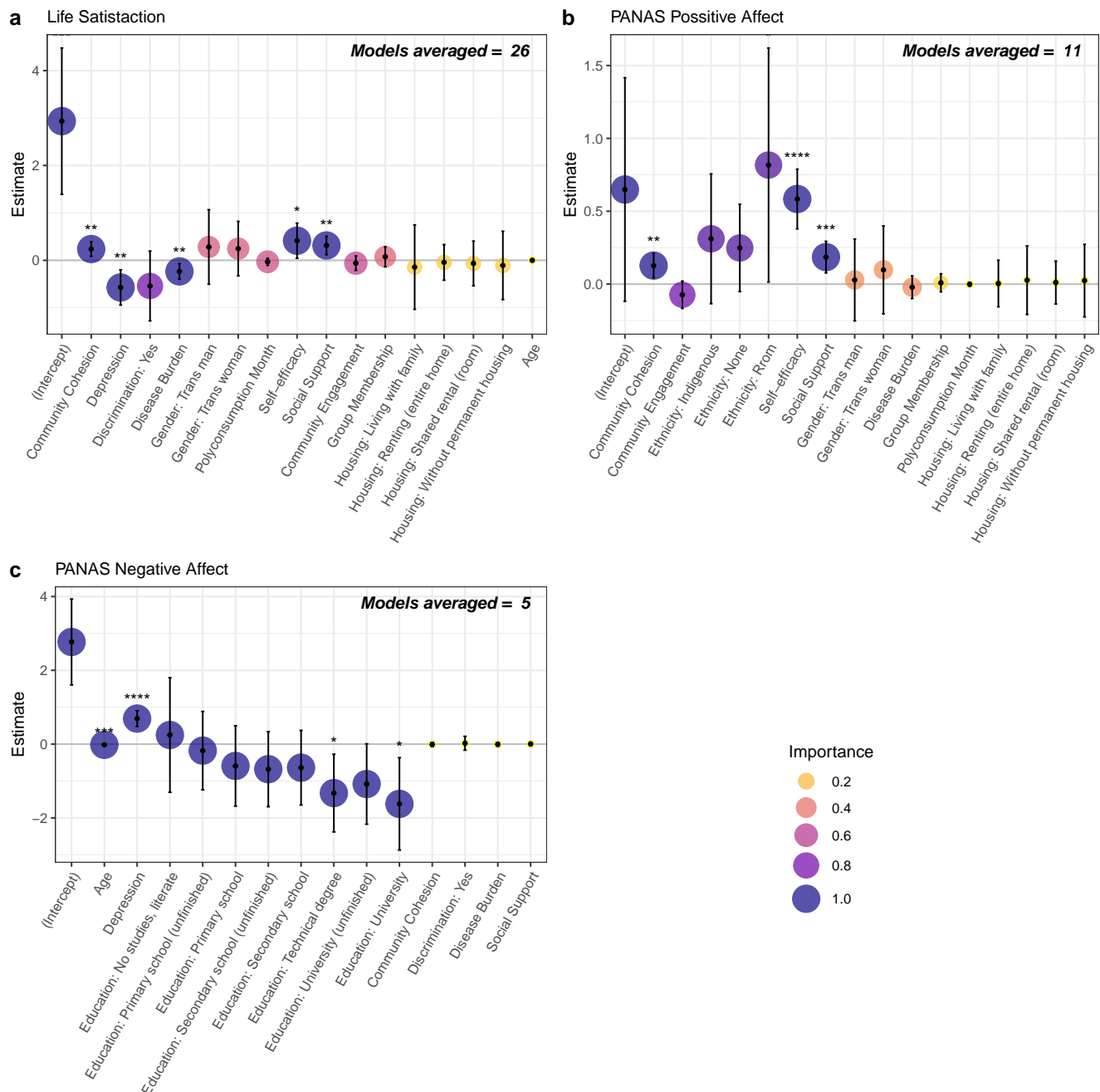
Figure 3. Term estimates with 95% confidence intervals for the final model averages of the best (AICc $\Delta < 2$) models predicting **(a)** Life Satisfaction, **(b)** PANAS Positive Affect, and **(c)** PANAS Negative Affect. Importance is the proportion of averaged models in which a term appears (i.e. an Importance of 0.5 means that the term is included in half the averaged models, and an Importance of 1.0 means that all the averaged models included that term). Larger and darker points indicate higher importance. Confidence intervals show uncertainty in effect size estimates. For term significance: † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 0.0001$.

6   Session info (for reproducibility)

```
# Display session information for reproducibility
# - Uses `pander()` for better formatting
```

```
# - `locale = FALSE` to exclude locale-specific info
library(pander)
pander(sessionInfo(), locale = FALSE)
```

R version 4.5.0 (2025-04-11)

Platform: x86_64-pc-linux-gnu

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

other attached packages: *pander(v.0.6.6)*, *pdftools(v.3.5.0)*, *ggplotify(v.0.1.2)*, *magick(v.2.8.6)*, *Hmisc(v.5.2-3)*, *gtsummary(v.2.2.0)*, *ggpubr(v.0.6.0)*, *lubridate(v.1.9.4)*, *forcats(v.1.0.0)*, *stringr(v.1.5.1)*, *dplyr(v.1.1.4)*, *purrr(v.1.0.4)*, *tidyr(v.1.3.1)*, *tibble(v.3.2.1)*, *ggplot2(v.3.5.2)*, *tidyverse(v.2.0.0)*, *car(v.3.1-3)*, *carData(v.3.0-5)*, *kableExtra(v.1.4.0)*, *scales(v.1.4.0)*, *readr(v.2.1.5)*, *performance(v.0.13.0)*, *MuMIn(v.1.48.11)*, *psych(v.2.5.3)*, *ltm(v.1.2-0)*, *polycor(v.0.8-1)*, *msm(v.1.8.2)*, *MASS(v.7.3-65)* and *knitr(v.1.50)*

loaded via a namespace (and not attached): *mnormt(v.2.1.1)*, *gridExtra(v.2.3)*, *rlang(v.1.1.6)*, *magrittr(v.2.0.3)*, *compiler(v.4.5.0)*, *systemfonts(v.1.2.3)*, *vctrs(v.0.6.5)*, *pkgconfig(v.2.0.3)*, *crayon(v.1.5.3)*, *fastmap(v.1.2.0)*, *backports(v.1.5.0)*, *labeling(v.0.4.3)*, *rmarkdown(v.2.29)*, *tzdb(v.0.5.0)*, *bit(v.4.6.0)*, *xfun(v.0.52)*, *broom(v.1.0.8)*, *parallel(v.4.5.0)*, *cluster(v.2.1.8.1)*, *R6(v.2.6.1)*, *stringi(v.1.8.7)*, *RColorBrewer(v.1.1-3)*, *rpart(v.4.1.24)*, *Rcpp(v.1.0.14)*, *bookdown(v.0.43)*, *base64enc(v.0.1-3)*, *Matrix(v.1.7-3)*, *splines(v.4.5.0)*, *nnet(v.7.3-20)*, *timechange(v.0.3.0)*, *tidyselect(v.1.2.1)*, *viridis(v.0.6.5)*, *rstudioapi(v.0.17.1)*, *abind(v.1.4-8)*, *yaml(v.2.3.10)*, *admisc(v.0.38)*, *qpdf(v.1.3.5)*, *lattice(v.0.22-7)*, *withr(v.3.0.2)*, *askpass(v.1.2.1)*, *evaluate(v.1.0.3)*, *foreign(v.0.8-90)*, *archive(v.1.1.12)*, *gridGraphics(v.0.5-1)*, *survival(v.3.8-3)*, *xml2(v.1.3.8)*, *pillar(v.1.10.2)*, *checkmate(v.2.3.2)*, *stats4(v.4.5.0)*, *insight(v.1.1.0)*, *generics(v.0.1.3)*, *vroom(v.1.6.5)*, *hms(v.1.1.3)*, *glue(v.1.8.0)*, *tools(v.4.5.0)*, *data.table(v.1.17.0)*, *ggsignif(v.0.6.4)*, *fs(v.1.6.6)*, *mvtnorm(v.1.3-3)*, *cowplot(v.1.1.3)*, *grid(v.4.5.0)*, *cards(v.0.6.0)*, *colorspace(v.2.1-1)*, *nlme(v.3.1-168)*, *htmlTable(v.2.4.3)*, *Formula(v.1.2-5)*, *cli(v.3.6.5)*, *textshaping(v.1.0.0)*, *expm(v.1.0-0)*, *viridisLite(v.0.4.2)*, *svglite(v.2.2.1)*, *gtable(v.0.3.6)*, *rstatix(v.0.7.2)*, *yulab.utils(v.0.2.0)*, *digest(v.0.6.37)*, *htmlwidgets(v.1.6.4)*, *farver(v.2.1.2)*, *htmltools(v.0.5.8.1)*, *lifecycle(v.1.0.4)* and *bit64(v.4.6.0-1)*

---

7   Supplementary references

Andresen, E. M., Malmgren, J. A., Carter, W. B., & Patrick, D. L. (1994). Screening for Depression in Well Older Adults: Evaluation of a Short Form of the CES-D. *American Journal of Preventive Medicine*, *10*(2), 77–84. https://doi.org/10.1016/S0749-3797(18)30622-6

Baessler, J., & Schwarzer, R. (1996). Evaluación de la autoeficacia: Adaptación española de la Escala de Autoeficacia general. [Measuring optimistic self-beliefs: A Spanish adaptation of the General Self-Efficacy Scale.] *Ansiedad y Estrés*, *2*(1), 1–8. https://psycnet.apa.org/record/1999-00958-001

Bartoń, K. (2025). *Mumin: Multi-model inference* [R package version 1.48.11]. https://doi.org/10.32614/CRAN.package.MuMIn

Diener, E., Emmons, R. A., Larsen, R. J., & Griffin, S. (1985). The Satisfaction With Life Scale. *Journal of Personality Assessment*, *49*(1), 71–75. https://doi.org/10.1207/s15327752jpa4901_13

Graham, J. W. (2012). *Missing Data: Analysis and design.* Springer. https://doi.org/10.1007/978-1-4614-4018-5

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

Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1–25. https://doi.org/10.18637/jss.v017.i05

Sherbourne, C. D., & Stewart, A. L. (1991). The MOS social support survey. *Social Science & Medicine*, *32*(6), 705–714. https://doi.org/10.1016/0277-9536(91)90150-B

Sinclair, V. G., & Wallston, K. A. (2004). The Development and Psychometric Evaluation of the Brief Resilient Coping Scale. *Assessment*, *11*(1), 94–101. https://doi.org/10.1177/1073191103258144

Stanton, M. C., Ali, S., & Chaudhuri, S. (2017). Individual, social and community-level predictors of wellbeing in a US sample of transgender and gender non-conforming individuals. *Culture, Health & Sexuality*, *19*(1), 32–49. https://doi.org/10.1080/13691058.2016.1189596

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, *54*(6), 1063–1070. https://doi.org/10.1037/0022-3514.54.6.1063

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis.* Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., . . . Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation* [R package version 1.1.3]. https://CRAN.R-project.org/package=dplyr

Xie, Y. (2014). Knitr: A comprehensive tool for reproducible research in R [ISBN 978-1466561595]. In V. Stodden, F. Leisch & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman and Hall/CRC. https://doi.org/10.1201/9781315373461-1

Zhu, H. (2021). *Kableextra: Construct complex table with 'kable' and pipe syntax* [R package version 1.3.4]. https://CRAN.R-project.org/package=kableExtra