Supplemental Material

This document explains how to reproduce the analyses included in the 2022 paper "Sibling Models Can test Causal Claims without Experiments: Applications for Pyschology" by S. Mason Garrison, Hannah Robertson, Jonathan D. Trattner, and Joseph Lee Rodgers.

If you are familiar with version-control using git and GitHub, the easiest way to replicate our analysis is by cloning the project repository hosted at github.com/R-Computing-Lab/target-causal-claims. Once cloned, you can open your favorite R IDE (e.g., RStudio or VSCode), and run the following:

```
# Install renv if needed
if (!require("renv")) install.packages("renv")
# Retrieve the packages used in the analysis
renv::restore()
# Trigger the targets pipeline
targets::tar_make()
```

This will recreate our analyses using the targets package, a pipeline toolkit for R by Will Landau (Landau 2021). Calling `targets::tar_make()` will trigger the pipeline, performing all data cleaning and modeling described in our paper. If you wish to replicate our analyses without cloning the repository from GitHub, please see the detailed walk-through for one of the studies below (the procedure is similar for all).

For this example, we are looking at whether neuroticism causally influences mental health. The first thing to do is load the packages we need for our analysis and define a utility function to reverse code survey items (necessary for combining measures from the TIPI).

```
library(tidyverse)
# Special installation of NlsyLinks needed
if (!require("NlsyLinks")) {
  install.packages(
  "NlsyLinks",
  repos = "https://mran.microsoft.com/snapshot/2021-04-01"
  )
}
library(NlsyLinks)
library(discord)
library(gtsummary)

#' Reverse Code Survey Items
#'
#' @description From https://github.com/svmiller/stevemisc/blob/master/R/revcode.R
#' @param x A vector of survey items
reverse_code <- function(x) {
  len <- length(na.omit(unique(x)))+1
  return((x*-1) + len)
}
```

The next step is to read in and process the data. Please see the code comments for specifics on individual steps.

```r
# Set the url for neuroticism data
neuroticism_data_url <- "https://raw.githubusercontent.com/R-Computing-Lab/target-causalclaims/main/data

# Read in the raw data
processed_neuroticism_data <- read_csv(
  neuroticism_data_url
) %>%
  # Select the variables, renaming them according to the fields from NLSY
  select(
    case_id = R0000100,
    sample_id = R0173600,
    race = R0214700,
    sex = R0214800,
    depression = H0013301, # H0013301 is depression at age 50
    anxious_upset = T4998603,
    calm_stable = T4998608
    ) %>%
  # Filter the data frame so we only have values greater than zero in the columns 'depression_score', '
    filter(
      if_all(
        .cols = c(depression, anxious_upset, calm_stable),
        function(value) value >= 0)
      ) %>%
  # Define neuroticism and scale relevant measures
    mutate(
      depression = scale(depression)[,1],
      # Reverse Coding calm/stable measurement from TIPI and adding with
      # anxious/upset
      neuroticism = scale(
        reverse_code(calm_stable) + anxious_upset
        )[,1]
      )
```

In order to perform the discordant kinship regression, we need to properly define familial relationships for NLSY subjects. This is easily done using the the NlsyLinks R package(Beasley et al. 2016), the methods of which are described in (Rodgers et al. 2016).

```r
single_entered_neuroticism <- processed_neuroticism_data %>%
  mutate(
    SubjectTag = case_id * 100
    ) %>%
    CreatePairLinksSingleEntered(
      # Get familial links from the NLSY79 dataset for full siblings
      linksPairDataset = Links79PairExpanded %>%
        filter(
          RelationshipPath == "Gen1Housemates" & RFull == 0.5
          ),
      linksNames = c(
        "ExtendedID",
        "R",
        "RelationshipPath"
        ),
      outcomeNames = c(
```

```
      "depression",
      "neuroticism",
      "sex",
      "race"
      ),
    subject1Qualifier = "_s1",
    subject2Qualifier = "_s2"
    ) %>%
  # NAs in outcome variable cannot be present for discordant regression to work
  drop_na(
    contains("depression")
    ) %>%
  as_tibble() %>%
  rename(
    subject_tag_s1 = SubjectTag_S1,
    subject_tag_s2 = SubjectTag_S2,
    extended_id = ExtendedID,
    relationship_path = RelationshipPath
  )
```

With this linked data, we can fit the discordant regression as follows:

```
neuroticism_fit <- discord_regression(
      data = single_entered_neuroticism,
      outcome = "depression",
      predictors = "neuroticism",
      sex = "sex",
      race = "race",
      pair_identifiers = c("_s1", "_s2")
    )
```

And view an aesthetically pleasing table of regression results:

```
neuroticism_fit %>%
 tbl_regression() %>%
    add_glance_source_note() %>%
    modify_header(
      statistic ~ "**T-Statistic**",
      p.value ~ "**P-Value**"
      )
```

Characteristic	Beta	T-Statistic	95% CI	P-Value
depression_mean	0.83	39.1	0.78, 0.87	<0.001
neuroticism_diff	0.06	4.94	0.03, 0.08	<0.001
neuroticism_mean	-0.03	-1.41	-0.07, 0.01	0.2
sex_1	0.04	1.26	-0.02, 0.10	0.2
race_1	-0.08	-3.94	-0.12, -0.04	<0.001
sex_2	-0.06	-2.10	-0.12, 0.00	0.036

Out GitHub repository contains both the data for the other questions addressed in our paper and commented code for the specific variable transformations.

Citations

Beasley, Will, Joe Rodgers, David Bard, Michael Hunter, S. Mason Garrison, and and Kelly Meredith. 2016. "NlsyLinks: Utilities and Kinship Information for Research with the NLSY." https://CRAN.R-project.org/package=NlsyLinks.

Landau, William Michael. 2021. "The Targets r Package: A Dynamic Make-Like Function-Oriented Pipeline Toolkit for Reproducibility and High-Performance Computing." *Journal of Open Source Software* 6 (57): 2959. https://doi.org/10.21105/joss.02959.

Rodgers, Joseph Lee, William H. Beasley, David E. Bard, Kelly M. Meredith, Michael D. Hunter, Amber B. Johnson, Maury Buster, et al. 2016. "The NLSY Kinship Links: Using the Nlsy79 and NLSY-Children Data to Conduct Genetically-Informed and Family-Oriented Research." *Behavior Genetics* 46 (4): 538–51. https://doi.org/10.1007/s10519-016-9785-3.