Moral Foundations Sacredness Scale: Measuring Morality

Jennifer Lin

1/6/2020

Contents

Introduction

The Measuring Morality Dataset contains questionnaires collected by researchers at Duke University.

From this dataset, I am interested in the Moral Foundations Sacredness Scale, adopted from the Moral Tradeoffs questionniare that was used in Study 3 of Graham, Haidt and Nosek's paper. While this is a shortened version of the one used in the original paper, it can still be insightful for the purposes of a replication to the results of the original study.

Before the analysis, I set up the process by loading some packages that will come in handy.

```r
# Load packages
library(tidyverse)
library(psych)
library(ggplot2)
library(GGally)
library("ggpubr")
library("reshape2")
library(scales)
library(lsr)
```

Moral Foundations Sacredness Scale

Clean Data

I begin by loading the data available here.

```
morals <- read.csv("~/Desktop/Working/Moral-Psychology/MMorality/mfss.csv",
    header = TRUE)
```

I create an average score for each foundation by averaging an individual's response to each of the questions that belong in each foundation.

```
### Harm ###
morals$Harm <- rowMeans(morals[, c("dogkick", "overweight", "palm")],
    na.rm = TRUE)

### Fairness ###
morals$Fairness <- rowMeans(morals[, c("cards", "ballots", "racepledge")],
    na.rm = TRUE)

### Ingroup ###
morals$Ingroup <- rowMeans(morals[, c("flagburn", "talkradio",
    "familyshun")], na.rm = TRUE)

### Authority ###
morals$Authority <- rowMeans(morals[, c("parentcurse", "handgesture",
    "rottentomato")], na.rm = TRUE)

### Purity ###
morals$Purity <- rowMeans(morals[, c("soulsell", "molesterblood",
    "stageanimal")], na.rm = TRUE)
```

For the purposes of the graph, I create an ordered factor variable for the political ideology question.

```
morals$ideology <- as.character(as.numeric(morals$ideo7))

morals$ideology <- recode(morals$ideology, `1` = "Extremely Liberal")
morals$ideology <- recode(morals$ideology, `2` = "Liberal")
morals$ideology <- recode(morals$ideology, `3` = "Slightly Liberal")
morals$ideology <- recode(morals$ideology, `4` = "Moderate")
morals$ideology <- recode(morals$ideology, `5` = "Slightly Conservative")
morals$ideology <- recode(morals$ideology, `6` = "Conservative")
morals$ideology <- recode(morals$ideology, `7` = "Extremely Conservative")
# Rid implicit NAs for the ideology variable
library(forcats)
```

```
morals$ideology <- fct_explicit_na(morals$ideology, na_level = "NA")

# Convert to Factor
morals$ideology <- as.factor(morals$ideology)
morals$ideology <- factor(morals$ideology, levels = c("Extremely Liberal",
    "Liberal", "Slightly Liberal", "Moderate", "Slightly Conservative",
    "Conservative", "Extremely Conservative"))

# Remove NA
library(forcats)
morals$ideology <- fct_explicit_na(morals$ideology, na_level = "NA")

# Remove NA from ideology
morals <- morals[!(morals$ideology == "NA"), ]

table(morals$ideology)
```

```
##
##      Extremely Liberal                Liberal       Slightly Liberal
##                     43                    192                    166
##               Moderate  Slightly Conservative           Conservative
##                    533                    199                    314
## Extremely Conservative                     NA
##                     54                      0
```

Descriptive Statistics Plot

To create the plot, I generate average scores for each foundation as a function of each level of political ideology.

```
Harm <- aggregate(Harm ~ ideology, morals, mean, na.rm = TRUE)
Fairness <- aggregate(Fairness ~ ideology, morals, mean, na.rm = TRUE)
Ingroup <- aggregate(Ingroup ~ ideology, morals, mean, na.rm = TRUE)
Authority <- aggregate(Authority ~ ideology, morals, mean, na.rm = TRUE)
Purity <- aggregate(Purity ~ ideology, morals, mean, na.rm = TRUE)
```

To generate a data frame that is usable to graph, I merge each of the data frames that were created above to one single frame.

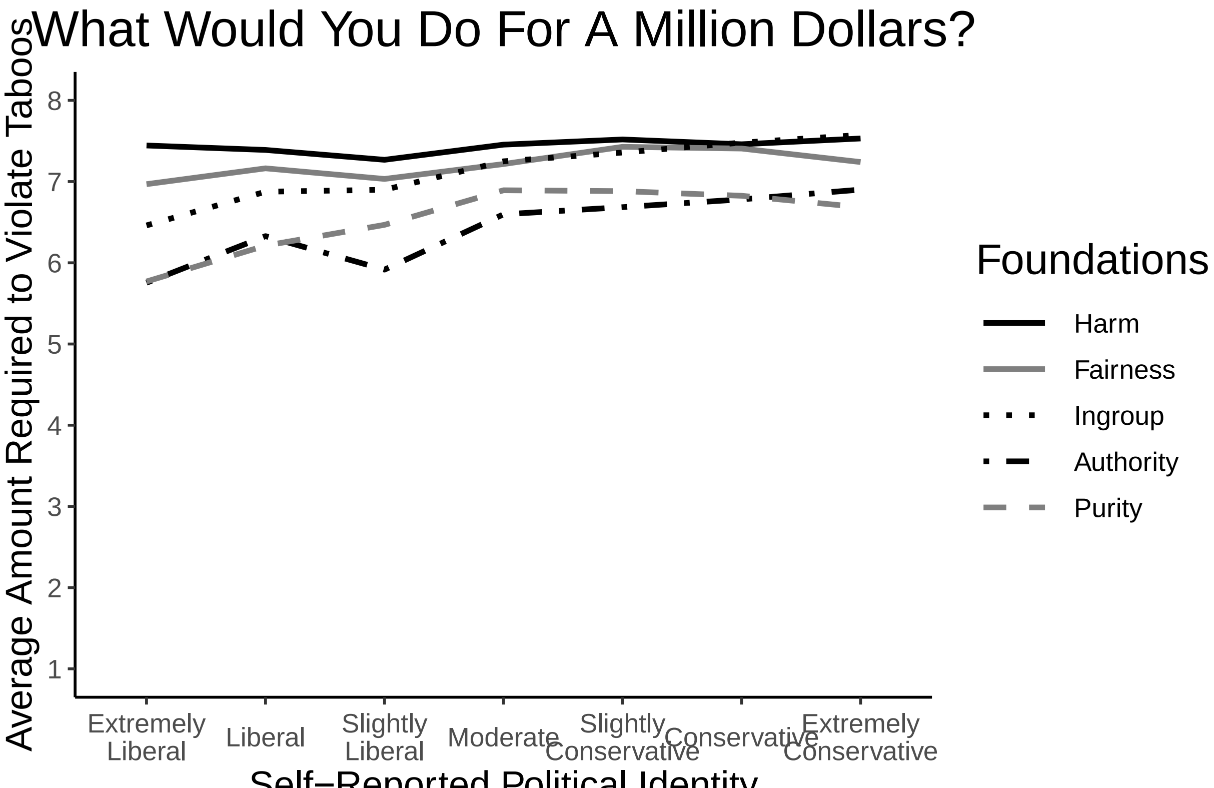
```
moral <- merge(Harm, Fairness, by.x = "ideology", by.y = "ideology",
    all.x = TRUE, all.y = TRUE)
moral <- merge(moral, Ingroup, by.x = "ideology", by.y = "ideology",
    all.x = TRUE, all.y = TRUE)
moral <- merge(moral, Authority, by.x = "ideology", by.y = "ideology",
    all.x = TRUE, all.y = TRUE)
```

```
moral <- merge(moral, Purity, by.x = "ideology", by.y = "ideology",
    all.x = TRUE, all.y = TRUE)

mfq <- reshape2::melt(moral, id.var = "ideology")
```

Now, I graph.

```
ggplot(mfq, aes(x = ideology, y = value, group = variable)) +
    geom_line(aes(linetype = variable, color = variable), size = 1) +
    theme_classic() + scale_linetype_manual("Foundations", breaks = c("Harm",
    "Fairness", "Ingroup", "Authority", "Purity"), values = c(Harm = "solid",
    Fairness = "solid", Ingroup = "dotted", Authority = "dotdash",
    Purity = "dashed")) + scale_color_manual("Foundations", breaks = c("Harm",
    "Fairness", "Ingroup", "Authority", "Purity"), values = c(Harm = "black",
    Fairness = "grey50", Ingroup = "black", Authority = "black",
    Purity = "grey50")) + ggtitle("What Would You Do For A Million Dollars?") +
    xlab("Self-Reported Political Identity") + ylab("Average Amount Required to Violate
    labs(caption = "Source: Graham, Haidt, and Nosek, 2009, Study 3") +
    theme(text = element_text(size = 12, colour = "black"), axis.title = element_text(s
        colour = "black"), title = element_text(size = 16, colour = "black"),
        plot.caption = element_text(size = 10, color = "black"),
        axis.text.x = element_text(angle = 0, hjust = 0.5, vjust = 0.5),
        plot.title = element_text(hjust = 0.5), legend.key.width = unit(2,
            "line")) + scale_x_discrete(labels = wrap_format(10)) +
    scale_y_continuous(breaks = seq(1, 8, 1), limits = c(1, 8))
```



Source: Graham, Haidt, and Nosek, 2009, Study 3

Cronbach's Alpha

I use the **psych** package to calculate the Cronbach's Alpha for the questions in each foundation

```
# Harm
Harm <- morals %>% select(c("dogkick", "overweight", "palm"))
psych::alpha(Harm)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Harm)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.66      0.67    0.58      0.41 2.1 0.014  7.4 1.1     0.42
##
##  lower alpha upper     95% confidence boundaries
## 0.63 0.66 0.69
##
##  Reliability if an item is dropped:
##            raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## dogkick         0.59      0.60    0.43      0.43 1.5    0.021    NA  0.43
## overweight      0.54      0.55    0.38      0.38 1.2    0.023    NA  0.38
## palm            0.56      0.59    0.42      0.42 1.4    0.021    NA  0.42
##
##  Item statistics
##               n raw.r std.r r.cor r.drop mean  sd
## dogkick    1475  0.71  0.77  0.57   0.47  7.7 1.1
## overweight 1481  0.83  0.79  0.62   0.51  7.2 1.7
## palm       1481  0.78  0.78  0.59   0.48  7.4 1.4
##
## Non missing response frequency for each item
##               1    2    3    4    5    6    7    8 miss
## dogkick    0.01 0.00 0.01 0.02 0.02 0.02 0.04 0.88 0.02
## overweight 0.02 0.01 0.03 0.04 0.05 0.04 0.07 0.74 0.01
## palm       0.03 0.00 0.01 0.02 0.02 0.05 0.09 0.79 0.01
```

```r
# Fairness
Fairness <- morals %>% select(c("cards", "ballots", "racepledge"))
psych::alpha(Fairness)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Fairness)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.72      0.73    0.64      0.47 2.7 0.012  7.3 1.3     0.48
##
##  lower alpha upper     95% confidence boundaries
## 0.7 0.72 0.74
##
##  Reliability if an item is dropped:
##            raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## cards           0.65      0.65    0.49      0.49 1.9    0.018    NA  0.49
## ballots         0.61      0.62    0.45      0.45 1.6    0.019    NA  0.45
## racepledge      0.64      0.65    0.48      0.48 1.9    0.018    NA  0.48
##
##  Item statistics
##               n raw.r std.r r.cor r.drop mean  sd
## cards      1481  0.84  0.80  0.63   0.54  6.8 1.9
## ballots    1479  0.79  0.82  0.67   0.57  7.5 1.4
## racepledge 1481  0.78  0.80  0.64   0.54  7.4 1.4
##
## Non missing response frequency for each item
##               1    2    3    4    5    6    7    8 miss
## cards      0.03 0.01 0.04 0.06 0.07 0.06 0.09 0.63 0.01
## ballots    0.02 0.00 0.01 0.02 0.03 0.04 0.06 0.82 0.01
## racepledge 0.03 0.00 0.01 0.02 0.02 0.05 0.09 0.79 0.01
```

```r
# Ingroup
Ingroup <- morals %>% select(c("flagburn", "talkradio", "familyshun"))
psych::alpha(Ingroup)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Ingroup)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.64      0.64    0.56      0.37 1.8 0.015  7.2 1.3     0.32
##
##  lower alpha upper     95% confidence boundaries
## 0.61 0.64 0.67
##
##  Reliability if an item is dropped:
##            raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## flagburn        0.48      0.49    0.32      0.32 0.96    0.026    NA  0.32
## talkradio       0.41      0.43    0.27      0.27 0.74    0.029    NA  0.27
## familyshun      0.68      0.69    0.52      0.52 2.20    0.016    NA  0.52
##
##  Item statistics
##               n raw.r std.r r.cor r.drop mean  sd
## flagburn   1476  0.83  0.78  0.63   0.50  7.1 1.9
## talkradio  1480  0.82  0.81  0.67   0.55  7.1 1.7
## familyshun 1483  0.64  0.70  0.42   0.34  7.5 1.3
##
## Non missing response frequency for each item
##               1    2    3    4    5    6    7    8 miss
## flagburn   0.04 0.01 0.02 0.05 0.03 0.04 0.08 0.72 0.02
## talkradio  0.02 0.01 0.02 0.05 0.05 0.06 0.09 0.70 0.01
## familyshun 0.02 0.00 0.00 0.01 0.02 0.04 0.13 0.76 0.01
```

```r
# Authority
Authority <- morals %>% select(c("parentcurse", "handgesture",
    "rottentomato"))
psych::alpha(Authority)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Authority)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.64      0.65    0.56      0.38 1.8 0.015  6.5 1.7     0.36
##
##  lower alpha upper     95% confidence boundaries
## 0.61 0.64 0.66
##
##  Reliability if an item is dropped:
##              raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## parentcurse       0.63      0.63    0.46      0.46 1.73    0.019    NA  0.46
## handgesture       0.43      0.48    0.32      0.32 0.93    0.025    NA  0.32
## rottentomato      0.48      0.53    0.36      0.36 1.11    0.024    NA  0.36
##
##  Item statistics
##                 n raw.r std.r r.cor r.drop mean  sd
## parentcurse  1477  0.63  0.73  0.48   0.39  7.4 1.5
## handgesture  1481  0.82  0.79  0.63   0.51  6.2 2.4
## rottentomato 1480  0.82  0.78  0.60   0.49  6.0 2.6
##
## Non missing response frequency for each item
##                 1    2    3    4    5    6    7    8 miss
## parentcurse  0.03 0.01 0.01 0.02 0.04 0.04 0.07 0.79 0.02
## handgesture  0.12 0.02 0.04 0.06 0.07 0.07 0.12 0.51 0.01
## rottentomato 0.14 0.02 0.04 0.06 0.08 0.06 0.09 0.51 0.01
```

```
# Purity
Purity <- morals %>% select(c("soulsell", "molesterblood", "stageanimal"))
psych::alpha(Purity)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Purity)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.43      0.51    0.42      0.26 1.1 0.023  6.7 1.5     0.22
##
##  lower alpha upper     95% confidence boundaries
## 0.39 0.43 0.48
##
##  Reliability if an item is dropped:
##               raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## soulsell           0.30      0.36    0.22      0.22 0.57    0.028    NA  0.22
## molesterblood      0.53      0.54    0.37      0.37 1.15    0.024    NA  0.37
## stageanimal        0.29      0.33    0.20      0.20 0.49    0.031    NA  0.20
##
##  Item statistics
##                  n raw.r std.r r.cor r.drop mean  sd
## soulsell      1476  0.63  0.73  0.51   0.31  7.3 1.7
## molesterblood 1480  0.82  0.66  0.35   0.25  5.5 3.0
## stageanimal   1483  0.62  0.74  0.54   0.34  7.2 1.4
```

```
##
## Non missing response frequency for each item
##                  1    2    3    4    5    6    7    8 miss
## soulsell      0.03 0.01 0.02 0.03 0.02 0.03 0.06 0.81 0.02
## molesterblood 0.26 0.01 0.02 0.04 0.05 0.04 0.08 0.50 0.01
## stageanimal   0.01 0.00 0.01 0.04 0.06 0.06 0.13 0.69 0.01
```

Repeated Measures GLM

To compare the aggregate individualizing and binding moral foundation, I generate the score below that represents an average of the responses to the questions under each category.

```r
### Individualizing and Binding items ###
morals$indiv <- rowMeans(morals[, c("dogkick", "overweight",
    "palm", "cards", "ballots", "racepledge")], na.rm = TRUE)
morals$bind <- rowMeans(morals[, c("flagburn", "talkradio", "familyshun",
    "parentcurse", "handgesture", "rottentomato", "soulsell",
    "molesterblood", "stageanimal")], na.rm = TRUE)
```

I then create a difference score that represents the difference between the indvidual and binding foundation scores. I conduct this analysis by running a linear regression with political ideology as a moderating variable.

```r
morals$diffscore <- morals$indiv - morals$bind

diff.model <- lm(diffscore ~ ideo7, data = morals)
summary(diff.model)
```

```
##
## Call:
## lm(formula = diffscore ~ ideo7, data = morals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9667 -0.5501 -0.2167  0.4434  5.1264
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.98147    0.07812  12.564  < 2e-16 ***
## ideo7       -0.10786    0.01753  -6.153 9.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9865 on 1486 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.02485,    Adjusted R-squared:  0.02419
## F-statistic: 37.86 on 1 and 1486 DF,  p-value: 9.745e-10
```

```
etaSquared(diff.model)
```

```
##           eta.sq eta.sq.part
## ideo7 0.02484611  0.02484611
```

The reported results are as follows:

- Aggregate difference between Indivdualizing and binding foundation: $F(1, 1486) = 157.854$, p < .001
- Moderation by Politics: $F(1, 1486) = 37.86$, p < .001, $\eta^2 = .025$