Moral Relevance: TAPS Wave 10

Jennifer Lin

11/25/2019

Contents

Introduction

The American Panel Study (data accessible here: https://wc.wustl.edu/taps-data-archive) contains the Moral Foundations Questionnaire in its 20-item version. This was administered to the panel in March of 2012, during Eave 10 of the sutdy. Here, I will use the 20-item questionnaire to create an analysis that replicates the work of Graham, Haidt and Nosek for the Moral Relevance subscale.

Before I begin, I load in the data and relevant packages.

```
# Load in data
taps = read.csv("~/Desktop/Working/Moral-Psychology/TAPS/TAPS10/taps10MFQ.csv",
    header = TRUE)
```

```
# Libraries
library(car)
library(dplyr)
library(psych)
library(ggplot2)
library(GGally)
library("ggpubr")
library("reshape2")
library(scales)
library(lme4)
library(lsr)
```

Clean Data

In this section, I organize the variables that I will need for the graph and the linear model.

First, I remove participants who did not pass the manipulation check items.

```
#### Remove Distractor Items ###

table(taps$attchecka)
```

```
##
##   1   2   3   4   5
## 812 372 219 165  83
```

```
taps <- taps[!(taps$attchecka == "4"), ]
taps <- taps[!(taps$attchecka == "5"), ]

table(taps$attcheckb)
```

```
##
##    1    2    3    4    5    6
##   27    6   17   54  160 1117
```

```
taps <- taps[!(taps$attcheckb == "1"), ]
taps <- taps[!(taps$attcheckb == "2"), ]
taps <- taps[!(taps$attcheckb == "3"), ]
```

Next, I create variables that represent aggregate scores on each of the moral foundations based on the Moral Relevance subscale.

```
### Harm ###
taps$emote <- taps$emote - 1
taps$weak <- taps$weak - 1
taps$Harm <- rowMeans(taps[, c("emote", "weak")], na.rm = TRUE)

### Fairness ###
taps$treatd <- taps$treatd - 1
taps$unfair <- taps$unfair - 1
taps$Fairness <- rowMeans(taps[, c("treatd", "unfair")], na.rm = TRUE)

### Ingroup ###
taps$lovec <- taps$lovec - 1
taps$betray <- taps$betray - 1
taps$Ingroup <- rowMeans(taps[, c("lovec", "betray")], na.rm = TRUE)

### Authority ###
taps$auth <- taps$auth - 1
```

```r
taps$conform <- taps$conform - 1
taps$Authority <- rowMeans(taps[, c("auth", "conform")], na.rm = TRUE)

### Purity ###
taps$purity <- taps$purity - 1
taps$purity <- taps$purity - 1
taps$Purity <- rowMeans(taps[, c("purity", "disgust")], na.rm = TRUE)
```

For the descriptive statistics line plot that I will create later, I recode a political ideology variable to reflect factor labels rather than numeric.

```r
table(taps$ideo7)
```

```
##
##   1   2   3   4   5   6   7
##  88 208 176 274 173 253  79
```

```r
# 1 = Very Liberal, 7 = very conservative
str(taps$ideo7)
```

```
##  int [1:2392] NA NA NA 2 3 4 NA NA 4 7 ...
```

```r
taps$ideology <- as.character(as.integer(taps$ideo7))
str(taps$ideology)
```

```
##  chr [1:2392] NA NA NA "2" "3" "4" NA NA "4" "7" "5" "6" NA "6" NA "6" ...
```

```r
taps$ideology <- recode(taps$ideology, `1` = "Very Liberal")
taps$ideology <- recode(taps$ideology, `2` = "Liberal")
taps$ideology <- recode(taps$ideology, `3` = "Slightly Liberal")
taps$ideology <- recode(taps$ideology, `4` = "Moderate")
taps$ideology <- recode(taps$ideology, `5` = "Slightly Conservative")
taps$ideology <- recode(taps$ideology, `6` = "Conservative")
taps$ideology <- recode(taps$ideology, `7` = "Very Conservative")

table(taps$ideology)
```

```
##
##          Conservative               Liberal              Moderate
##                   253                   208                   274
## Slightly Conservative      Slightly Liberal     Very Conservative
##                   173                   176                    79
##          Very Liberal
##                    88
```

```r
# Rid implicit NAs for the ideology variable
library(forcats)
taps$ideology <- fct_explicit_na(taps$ideology, na_level = "NA")
```

```r
table(taps$ideology)
```

```
##
##          Conservative               Liberal              Moderate
##                   253                   208                   274
## Slightly Conservative      Slightly Liberal     Very Conservative
##                   173                   176                    79
##          Very Liberal                    NA
##                    88                  1141
```

```r
# Establish factor order for graphing
taps$ideology <- as.factor(as.character(taps$ideology))
taps$ideology <- factor(taps$ideology, levels = c("Very Liberal",
    "Liberal", "Slightly Liberal", "Moderate", "Slightly Conservative",
    "Conservative", "Very Conservative"))
```

Cronbach's Alpha Calculation

I calculate the Cronbach's Alpha for each foundation on the Moral Relevance subscale

```r
# Harm
Harm2 <- taps %>% select(c("emote", "weak"))
psych::alpha(Harm2)
```

```
## Warning in matrix(unlist(drop.item), ncol = 10, byrow = TRUE): data length
## [16] is not a sub-multiple or multiple of the number of columns [10]
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Harm2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean sd median_r
##       0.68      0.68    0.52      0.52 2.1 0.013  2.8  1     0.52
##
##  lower alpha upper     95% confidence boundaries
## 0.65 0.68 0.71
##
##  Reliability if an item is dropped:
##       raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## emote      0.52      0.52    0.27      0.52  NA       NA  0.52  0.52
## weak       0.27      0.52      NA        NA  NA       NA  0.27  0.52
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
```

```
## emote 1325  0.88  0.87  0.63   0.52  2.7 1.2
## weak  1325  0.86  0.87  0.63   0.52  2.9 1.1
##
## Non missing response frequency for each item
##          0    1    2    3    4 miss
## emote 0.07 0.10 0.21 0.32 0.30 0.45
## weak  0.04 0.07 0.19 0.32 0.37 0.45
```

```
# Fairness
Fairness2 <- taps %>% select(c("treatd", "unfair"))
psych::alpha(Fairness2)
```

```
## Warning in matrix(unlist(drop.item), ncol = 10, byrow = TRUE): data length
## [16] is not a sub-multiple or multiple of the number of columns [10]
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Fairness2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.72      0.73    0.57      0.57 2.7 0.011  3.2 0.93     0.57
##
##  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
## treatd      0.57      0.57    0.33      0.57  NA       NA  0.57  0.57
## unfair      0.33      0.57      NA        NA  NA       NA  0.33  0.57
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean   sd
## treatd 1326  0.90  0.89  0.67   0.57  3.1 1.13
## unfair 1324  0.87  0.89  0.67   0.57  3.3 0.97
##
## Non missing response frequency for each item
##           0    1    2    3    4 miss
## treatd 0.05 0.06 0.14 0.28 0.47 0.45
## unfair 0.03 0.03 0.10 0.27 0.57 0.45
```

```
# Ingroup
Ingroup2 <- taps %>% select(c("lovec", "betray"))
psych::alpha(Ingroup2)
```

```
## Warning in matrix(unlist(drop.item), ncol = 10, byrow = TRUE): data length
## [16] is not a sub-multiple or multiple of the number of columns [10]
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Ingroup2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.64      0.64    0.47      0.47 1.8 0.015  2.5 1.1     0.47
##
##  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
## lovec       0.47      0.47    0.22      0.47  NA       NA  0.47  0.47
## betray      0.22      0.47      NA        NA  NA       NA  0.22  0.47
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## lovec  1327  0.87  0.86  0.59   0.47  2.4 1.3
## betray 1323  0.85  0.86  0.59   0.47  2.7 1.2
##
## Non missing response frequency for each item
##           0    1    2    3    4 miss
## lovec  0.11 0.16 0.22 0.25 0.26 0.45
## betray 0.07 0.12 0.20 0.28 0.33 0.45
```

```r
# Authority
Authority2 <- taps %>% select(c("auth", "conform"))
psych::alpha(Authority2)
```

```
## Warning in matrix(unlist(drop.item), ncol = 10, byrow = TRUE): data length
## [16] is not a sub-multiple or multiple of the number of columns [10]
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Authority2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean sd median_r
##       0.58      0.58    0.41      0.41 1.4 0.017  2.3  1     0.41
##
##  lower alpha upper     95% confidence boundaries
## 0.55 0.58 0.62
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## auth         0.41      0.41    0.17      0.41  NA       NA  0.41  0.41
## conform      0.17      0.41      NA        NA  NA       NA  0.17  0.41
```

```
##
##  Item statistics
##            n raw.r std.r r.cor r.drop mean  sd
## auth    1330  0.85  0.84  0.54   0.41  2.7 1.2
## conform 1325  0.83  0.84  0.54   0.41  1.9 1.2
##
## Non missing response frequency for each item
##            0    1    2    3    4 miss
## auth    0.06 0.11 0.20 0.29 0.34 0.44
## conform 0.13 0.25 0.29 0.24 0.09 0.45
```

```
# Purity
Purity2 <- taps %>% select(c("purity", "disgust"))
psych::alpha(Purity2)
```

```
## Warning in matrix(unlist(drop.item), ncol = 10, byrow = TRUE): data length
## [16] is not a sub-multiple or multiple of the number of columns [10]
```

```
##
## Reliability analysis
## Call: psych::alpha(x = Purity2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean sd median_r
##       0.61      0.61    0.44      0.44 1.6 0.016  2.8  1     0.44
##
##  lower alpha upper     95% confidence boundaries
## 0.58 0.61 0.64
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## purity       0.44      0.44     0.2      0.44  NA       NA  0.44  0.44
## disgust      0.20      0.44      NA        NA  NA       NA  0.20  0.44
##
##  Item statistics
##            n raw.r std.r r.cor r.drop mean  sd
## purity  1324  0.83  0.85  0.57   0.44  2.0 1.1
## disgust 1324  0.87  0.85  0.57   0.44  3.6 1.3
##
## Non missing response frequency for each item
##           -1    0    1    2    3    4    5 miss
## purity  0.04 0.07 0.16 0.28 0.44 0.00 0.00 0.45
## disgust 0.00 0.00 0.08 0.14 0.22 0.27 0.29 0.45
```

Lineplot – Descriptive Statistics

In this section, I will create a linegraph that displays the average score on each foundation as a function of the respondent's political ideology. To do this, I generate average scores by moral foundation.

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

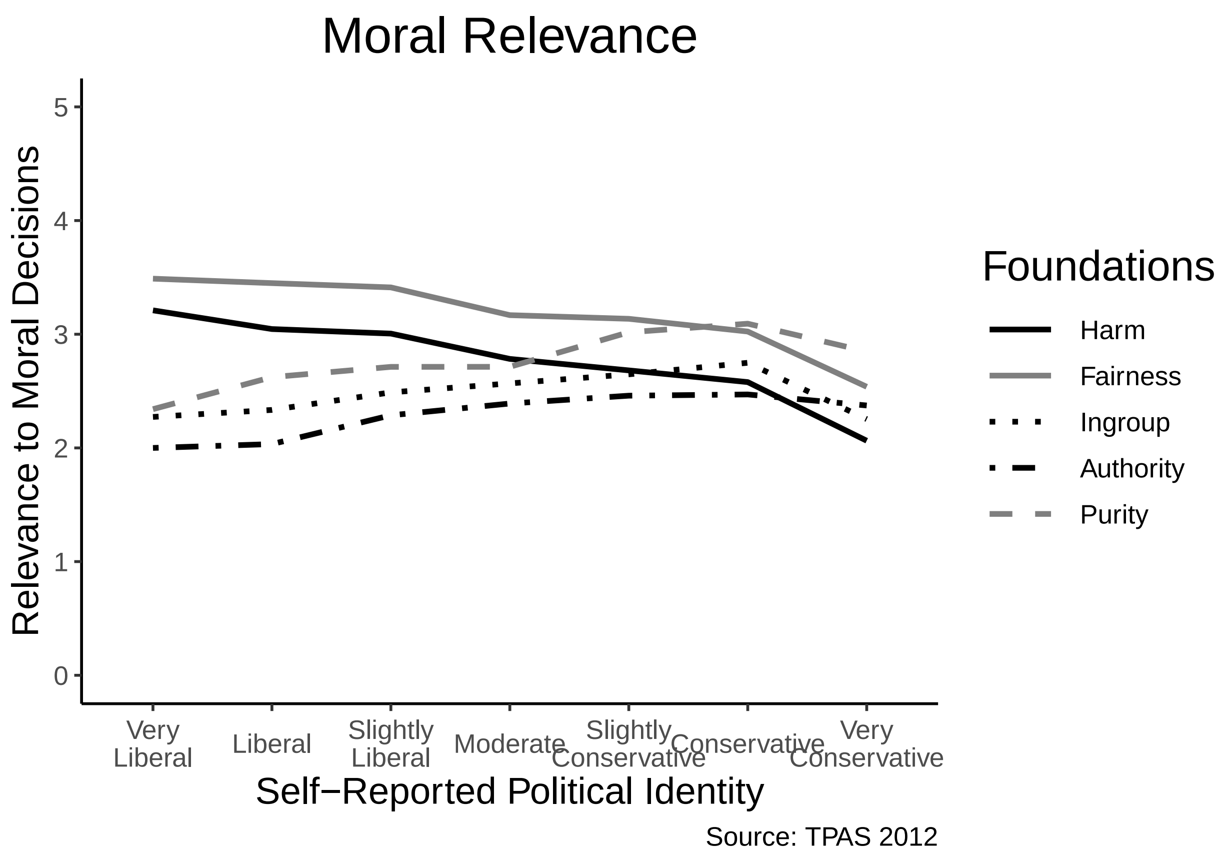
In order for ggplot to graph the data, the points need to be merged into one large data frame and reshaped into the proper data frame formation. I do this with the code below.

```
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 create the 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("Moral Relevance") + xlab("Self-Reported Political Ide
    ylab("Relevance to Moral Decisions") + ylim(0, 5) + labs(caption = "Source: TPAS 201
    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))
```



Repeated Measures GLM

To see if liberals and conservatives differ significantly on the individualizing versus binding foundations, the authors generated a repeated measures GLM to capture the relationship. Additionally, they tested to see if the relationship would be moderated by politics.

Here, I replicate the model using the TAPS dataset.

I generate a composite score for the individualizing and binding foundations.

```
# Individualizing and Binding scores
taps$indiv <- rowMeans(taps[, c("emote", "weak", "treatd", "unfair")],
    na.rm = TRUE)
taps$bind <- rowMeans(taps[, c("lovec", "betray", "auth", "conform",
    "purity", "disgust")], na.rm = TRUE)
```

Next, I generate a difference score between the individualizing and binding foundations.

```
taps$diffscore <- taps$indiv - taps$bind
```

Now, I run the model and print out a table summarizing the results and include an η^2 statistic.

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

```
##
## Call:
## lm(formula = diffscore ~ ideo7, data = taps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67649 -0.52571 -0.01581  0.49017  2.70241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.35514    0.05787   23.42   <2e-16 ***
## ideo7       -0.22422    0.01316  -17.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7997 on 1249 degrees of freedom
##   (1141 observations deleted due to missingness)
## Multiple R-squared:  0.1886, Adjusted R-squared:  0.188
## F-statistic: 290.3 on 1 and 1249 DF,  p-value: < 2.2e-16
```

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

```
##          eta.sq eta.sq.part
## ideo7 0.1886143   0.1886143
```

The results are interpreted as follows:

The `F-statistic: 288.7 on 1 and 1249 DF,  p-value: < 2.2e-16` reflects the moderation of politics in the model. To find the difference between the scales as is, we square the t-value next to the `(Intercept)` row and use that p-value

The results are as follows: - Aggregate difference between individualizing and binding foundations: F(1, 1249) = 546.1569, p < .001 - Moderation by politics: F(1, 1249) = 288.7, p < .001, η^2 = .188

To see the distribution of the scores, I generate a scatterplot with the linear model fitted. the political ideology variable is acress the x-axis and it is represented by 1 = Very Liberal to 7 = Very Conservative

```r
# Fit plot with linear regression line
ggplot(taps, aes(x = ideo7, y = diffscore)) + geom_point(size = 2) +
    geom_smooth(method = "lm", se = TRUE, fullrange = FALSE,
        level = 0.95) + theme_classic() + ggtitle("Moral Relevance") +
    xlab("Self-Reported Political Ideology") + ylab("Difference Score") +
    labs(caption = "Source: TAPS Wave 10") + theme(text = element_text(size = 12,
```

```
    colour = "black"), axis.title = element_text(size = 14, colour = "black"),
    title = element_text(size = 16, colour = "black"), plot.caption = element_text(size
        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"))
```