Moral Judgment: CCES 2012

Jennifer Lin

1/6/2020

Contents

Introduction

The CCES 2012 team module from Duke University contains the 30 item Moral Foundation Questionnaire. It has 739 observatiions for the dataset.

Here, I will recreate the moral judgment analyses used in the original paper but using the Duke CCES dataset.

Before I begin, let's load in the data (available for download here) and relevant packages.

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

```r
cces = read.csv("~/Desktop/Working/Moral-Psychology/DukeCCES12/CCES-MFQ.csv",
    header = TRUE)
```

Moral Foundations Questionnaire – 30-item

Clean Data

First, I create an analysis that uses the entirety of the Moral Foundations Questionnaire. I prepare the data below.

```
cces <- cces[!(cces$math == "4"), ]
cces <- cces[!(cces$math == "5"), ]
cces <- cces[!(cces$math == "6"), ]

cces <- cces[!(cces$dogood == "1"), ]
cces <- cces[!(cces$dogood == "3"), ]
```

```
### Harm ###
cces$compat <- cces$compat - 1
cces$hurt <- cces$hurt - 1
cces$kill <- cces$kill - 1
cces$Harm <- rowMeans(cces[, c("compat", "hurt", "kill")], na.rm = TRUE)

### Fairness ###
cces$treatf <- cces$treatf - 1
cces$justice <- cces$justice - 1
cces$rich <- cces$rich - 1
cces$Fairness <- rowMeans(cces[, c("treatf", "justice", "rich")],
    na.rm = TRUE)

### Ingroup ###
cces$proudc <- cces$proudc - 1
cces$loyalf <- cces$loyalf - 1
cces$team <- cces$team - 1
cces$Ingroup <- rowMeans(cces[, c("proudc", "loyalf", "team")],
    na.rm = TRUE)

### Authority ###
cces$authc <- cces$authc - 1
cces$roles <- cces$roles - 1
cces$obey <- cces$obey - 1
cces$Authority <- rowMeans(cces[, c("authc", "roles", "obey")],
    na.rm = TRUE)

### Purity ###
cces$nodisgust <- cces$nodisgust - 1
cces$notnat <- cces$notnat - 1
cces$chaste <- cces$chaste - 1
```

```
cces$Purity <- rowMeans(cces[, c("disgust", "notnat", "chaste")],
    na.rm = TRUE)
```

For this graph, we will make a lineplot that has political ideology on the x-axis and the average response score for the foundation on the y-axis. There will be a different line for each of the foundations.

I begin this process by recoding the political ideology variable

```
cces$ideology <- as.character(as.integer(cces$ideo5))

cces$ideology <- recode(cces$ideology, `1` = "Very Liberal")
cces$ideology <- recode(cces$ideology, `2` = "Liberal")
cces$ideology <- recode(cces$ideology, `3` = "Moderate")
cces$ideology <- recode(cces$ideology, `4` = "Conservative")
cces$ideology <- recode(cces$ideology, `5` = "Very Conservative")


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


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

```
##
##      Very Liberal           Liberal          Moderate      Conservative
##                22                40                59                55
## Very Conservative
##                30
```

Descriptive Statistics Plot

I create aggregate scores for each foundation as a function of political ideology

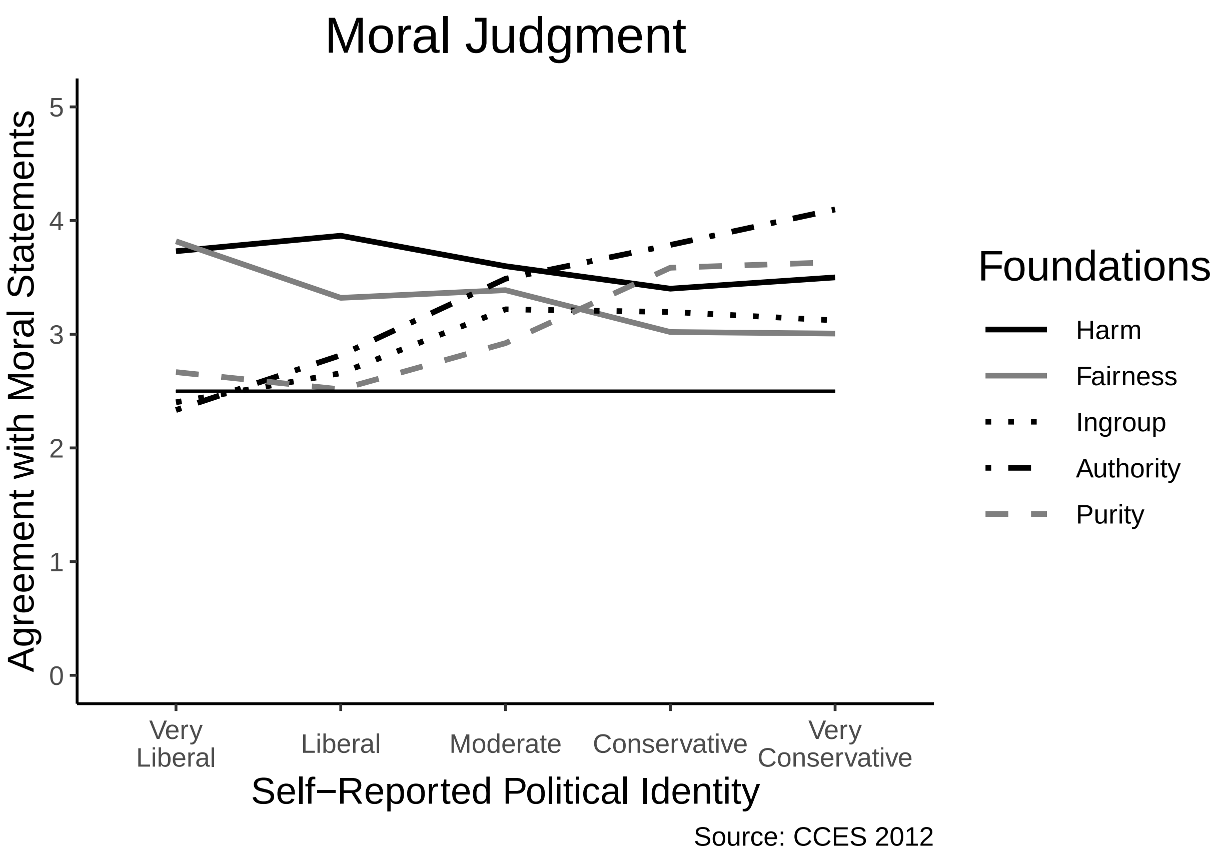
```
Harm <- aggregate(Harm ~ ideology, cces, mean, na.rm = TRUE)
Fairness <- aggregate(Fairness ~ ideology, cces, mean, na.rm = TRUE)
Ingroup <- aggregate(Ingroup ~ ideology, cces, mean, na.rm = TRUE)
Authority <- aggregate(Authority ~ ideology, cces, mean, na.rm = TRUE)
Purity <- aggregate(Purity ~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

The above method creates 5 different data frames. To create a graph, we need everything to be on one data frame. Here, I will merge each of these data frames into one and reshape them into a format that ggplot will be able to read.

```
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")
```

Finally, we plot!

```
ggplot(mfq, aes(x = ideology, y = value, group = variable)) +
    geom_line(aes(linetype = variable, color = variable), size = 1) +
    theme_classic() + geom_line(aes(y = 2.5)) + 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 Judgment") +
    xlab("Self-Reported Political Identity") + ylab("Agreement with Moral Statements") +
    ylim(0, 5) + labs(caption = "Source: CCES 2012") + theme(text = element_text(size =
    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")) + scale_x_discrete(labels = wrap_format(10))
```



Cronbach's Alpha

I calculate the Crombach's Alpha score for each foundation on the 30-item scale.

```r
# Harm
Harm <- cces %>% select(c("compat", "hurt", "kill"))
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
##        0.3      0.34    0.28      0.15 0.51 0.043  3.6 1.3     0.17
##
##  lower alpha upper     95% confidence boundaries
## 0.22 0.3 0.39
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r   S/N alpha se var.r  med.r
## compat     0.013     0.014  0.0071    0.0071 0.014    0.068    NA 0.0071
## hurt       0.372     0.411  0.2586    0.2586 0.698    0.041    NA 0.2586
```

```
## kill       0.291     0.294  0.1720    0.1720 0.415    0.052    NA 0.1720
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## compat 139  0.75  0.73  0.51  0.309  3.9 1.1
## hurt   131  0.69  0.60  0.20  0.086  4.2 1.3
## kill   121  0.78  0.64  0.32  0.162  2.7 1.9
##
## Non missing response frequency for each item
##           0    1    2    3    4    5 miss
## compat 0.02 0.02 0.04 0.23 0.32 0.37 0.81
## hurt   0.03 0.05 0.02 0.08 0.23 0.58 0.82
## kill   0.18 0.15 0.15 0.10 0.15 0.27 0.84
```

```
# Fairness
Fairness <- cces %>% select(c("treatf", "justice", "rich"))
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
##       0.18      0.21    0.16     0.079 0.26 0.05  3.3 1.1     0.14
##
##  lower alpha upper     95% confidence boundaries
## 0.08 0.18 0.28
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r    S/N alpha se var.r  med.r
## treatf     -0.094     -0.11  -0.052    -0.052 -0.099    0.070    NA -0.052
## justice     0.237      0.26   0.148     0.148  0.347    0.051    NA  0.148
## rich        0.247      0.25   0.142     0.142  0.330    0.055    NA  0.142
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean   sd
## treatf  130  0.62  0.69  0.44  0.204  4.2 1.06
## justice 131  0.50  0.58  0.15  0.031  4.2 0.95
## rich    140  0.66  0.59  0.16  0.070  1.6 1.64
##
## Non missing response frequency for each item
##            0    1    2    3    4    5 miss
## treatf  0.02 0.02 0.03 0.12 0.28 0.53 0.82
## justice 0.01 0.01 0.05 0.12 0.39 0.43 0.82
## rich    0.39 0.13 0.18 0.16 0.05 0.09 0.81
```

```r
# Ingroup
Ingroup <- cces %>% select(c("proudc", "loyalf", "team"))
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
##       0.42      0.41    0.33      0.19 0.7 0.036    3 1.3     0.15
##
##  lower alpha upper     95% confidence boundaries
## 0.35 0.42 0.49
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## proudc      0.46      0.47    0.30      0.30 0.87    0.039    NA  0.30
## loyalf      0.20      0.20    0.11      0.11 0.25    0.058    NA  0.11
## team        0.26      0.27    0.15      0.15 0.36    0.053    NA  0.15
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## proudc 147  0.67  0.62  0.25   0.16  4.0 1.2
## loyalf 129  0.81  0.72  0.48   0.32  2.8 1.6
## team   125  0.76  0.70  0.44   0.29  2.1 1.5
##
## Non missing response frequency for each item
##           0    1    2    3    4    5 miss
## proudc 0.02 0.03 0.08 0.16 0.26 0.46 0.80
## loyalf 0.10 0.12 0.21 0.18 0.20 0.19 0.83
## team   0.18 0.18 0.26 0.20 0.11 0.06 0.83
```

```r
# Authority
Authority <- cces %>% select(c("authc", "roles", "obey"))
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
##        0.5      0.51    0.43      0.26 1.1 0.03  3.4 1.3      0.3
##
##  lower alpha upper     95% confidence boundaries
## 0.45 0.5 0.56
```

```
##
##  Reliability if an item is dropped:
##       raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## authc      0.50      0.51    0.34      0.34 1.02    0.036    NA  0.34
## roles      0.44      0.46    0.30      0.30 0.86    0.038    NA  0.30
## obey       0.22      0.25    0.14      0.14 0.33    0.050    NA  0.14
##
##  Item statistics
##         n raw.r std.r r.cor r.drop mean  sd
## authc 127  0.64  0.67  0.38   0.26  4.3 1.1
## roles 135  0.80  0.69  0.43   0.32  2.7 1.8
## obey  134  0.82  0.77  0.59   0.42  3.2 1.6
##
## Non missing response frequency for each item
##          0    1    2    3    4    5 miss
## authc 0.01 0.02 0.03 0.13 0.19 0.61 0.83
## roles 0.19 0.12 0.08 0.22 0.19 0.20 0.82
## obey  0.10 0.04 0.15 0.22 0.24 0.25 0.82
```

```r
# Purity
Purity <- cces %>% select(c("disgust", "notnat", "chaste"))
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
##       0.71      0.71    0.71      0.45 2.4 0.018  3.1 1.4     0.37
##
##  lower alpha upper     95% confidence boundaries
## 0.68 0.71 0.75
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## disgust      0.86      0.86    0.75      0.75 6.14    0.010    NA  0.75
## notnat       0.36      0.36    0.22      0.22 0.56    0.047    NA  0.22
## chaste       0.54      0.54    0.37      0.37 1.17    0.034    NA  0.37
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## disgust 124  0.71  0.67  0.36   0.31  3.8 1.5
## notnat  118  0.89  0.89  0.87   0.73  2.7 1.7
## chaste  134  0.85  0.83  0.77   0.60  2.8 1.6
##
```

```
## Non missing response frequency for each item
##            0    1    2    3    4    5    6 miss
## disgust 0.00 0.10 0.13 0.22 0.19 0.23 0.15 0.83
## notnat  0.17 0.10 0.14 0.25 0.17 0.18 0.00 0.84
## chaste  0.13 0.13 0.11 0.24 0.21 0.19 0.00 0.82
```

Repeated Measures GLM

In this section, I create a linear model that attempts to replicate the results of the original paper. There, they aggregate the individualizing and binding foundation scores to test for differences between the foundation and use politics as a moderating variable.

Here, I use a linear model (which successfully replicated on the moral relevance side of the data on the original datasets) to test this relationship.

To begin, I create a difference score between the individualizing and binding foundations

```r
# Individualizing and Binding scores
cces$indiv <- rowMeans(cces[, c("compat", "hurt", "kill", "treatf",
    "justice", "rich")], na.rm = TRUE)
cces$bind <- rowMeans(cces[, c("proudc", "loyalf", "team", "authc",
    "roles", "obey", "disgust", "notnat", "chaste")], na.rm = TRUE)

# Create a difference score
cces$diffscore <- cces$indiv - cces$bind

# Run the linear model
diff.model <- lm(diffscore ~ ideo5, data = cces)
summary(diff.model)
```

```
##
## Call:
## lm(formula = diffscore ~ ideo5, data = cces)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9371 -0.8223  0.0357  0.6636  3.4620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8357     0.2196   8.361 9.63e-15 ***
## ideo5        -0.4996     0.0651  -7.673 6.78e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.124 on 204 degrees of freedom
##   (533 observations deleted due to missingness)
## Multiple R-squared:  0.224,  Adjusted R-squared:  0.2202
## F-statistic: 58.88 on 1 and 204 DF,  p-value: 6.778e-13
```

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

```
##          eta.sq eta.sq.part
## ideo5 0.2239874   0.2239874
```

The reported results are as follows:

- Aggregate difference between Indivdualizing and binding foundation: $F(1, 204) = 69.91$, p $< .001$
- Moderation by Politics: $F(1, 204) = 58.88$, p $< .001$, $\eta^2 = .23$

Moral Foundations Questionnaire – 20-item

In the following section, I repeat the analyses above with only the 20-item Moral Foundations Questionnaire. I am curious to see any differences in results that come from differences in the questionnaire used.

Before I begin with the linegraph and analyses, I prepare the data below.

Clean Data

```r
### Harm ###
cces$compat <- cces$compat - 1
cces$hurt <- cces$hurt - 1
cces$Harm <- rowMeans(cces[, c("compat", "hurt")], na.rm = TRUE)

### Fairness ###
cces$treatf <- cces$treatf - 1
cces$justice <- cces$justice - 1
cces$Fairness <- rowMeans(cces[, c("treatf", "justice")], na.rm = TRUE)

### Ingroup ###
cces$proudc <- cces$proudc - 1
cces$loyalf <- cces$loyalf - 1
cces$Ingroup <- rowMeans(cces[, c("proudc", "loyalf")], na.rm = TRUE)

### Authority ###
cces$authc <- cces$authc - 1
cces$roles <- cces$roles - 1
```

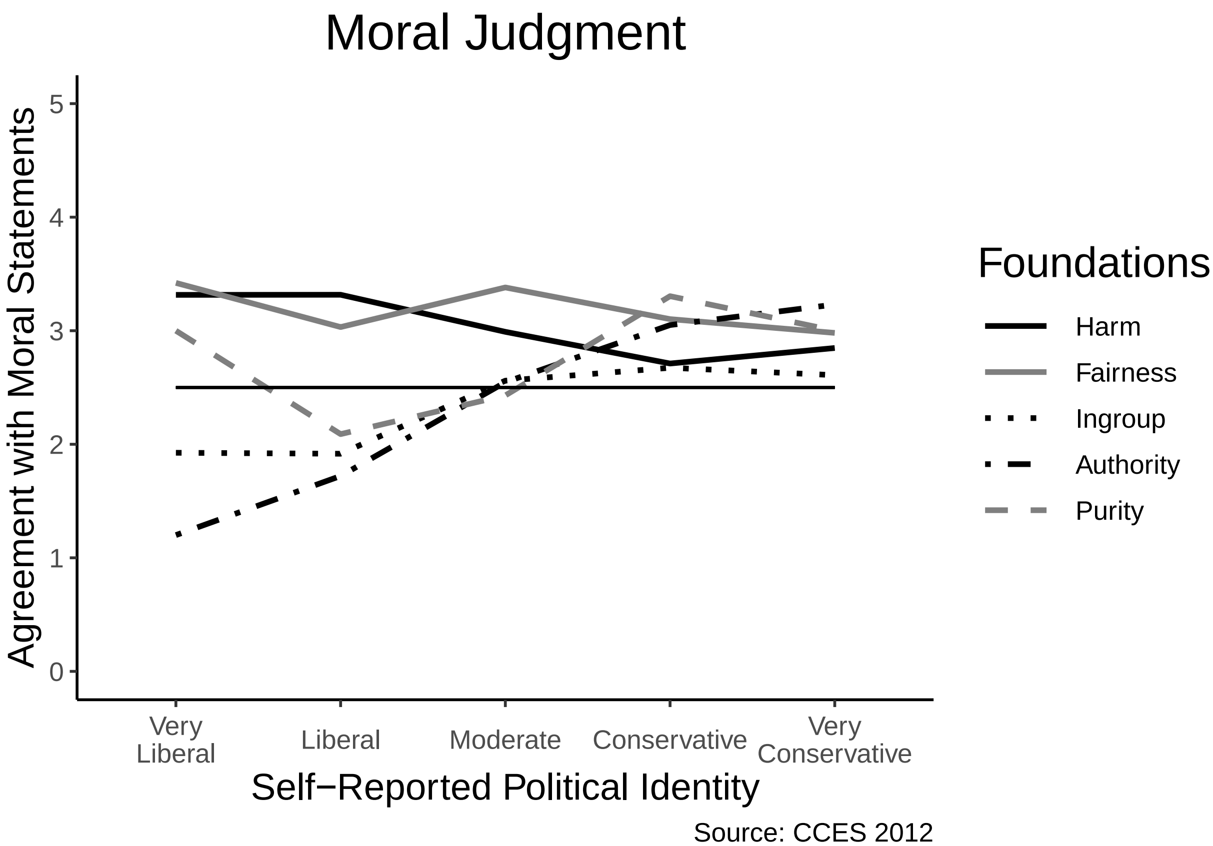
```r
cces$Authority <- rowMeans(cces[, c("authc", "roles")], na.rm = TRUE)

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

Descriptive Statistics Plot

```r
############# Aggregate By Ideology ##############

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

# Merge each of the aggregates into one large data frame to
# create plot using one frame rather than 5

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")
```

```r
ggplot(mfq, aes(x = ideology, y = value, group = variable)) +
    geom_line(aes(linetype = variable, color = variable), size = 1) +
    theme_classic() + geom_line(aes(y = 2.5)) + 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 Judgment") +
    xlab("Self-Reported Political Identity") + ylab("Agreement with Moral Statements") +
    ylim(0, 5) + labs(caption = "Source: CCES 2012") + theme(text = element_text(size =
    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")) + scale_x_discrete(labels = wrap_format(10))
```



Cronbach's Alpha

I calculate the Crombach's Alpha score for each foundation on the 20-item scale.

```
# Harm
Harm <- cces %>% select(c("compat", "hurt"))
psych::alpha(Harm)
```

```
## 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 = Harm)
##
##   raw_alpha std.alpha G6(smc) average_r  S/N   ase mean  sd median_r
##       0.29      0.29    0.17      0.17 0.42 0.052    3 1.2     0.17
```

```
##
##  lower alpha upper     95% confidence boundaries
## 0.19 0.29 0.39
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## compat      0.17      0.17    0.03      0.17  NA       NA  0.17  0.17
## hurt        0.03      0.17      NA        NA  NA       NA  0.03  0.17
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## compat 139  0.87  0.77  0.32   0.17  2.9 1.1
## hurt   131  0.89  0.77  0.32   0.17  3.2 1.3
##
## Non missing response frequency for each item
##          -1    0    1    2    3    4 miss
## compat 0.02 0.02 0.04 0.23 0.32 0.37 0.81
## hurt   0.03 0.05 0.02 0.08 0.23 0.58 0.82
```

```r
# Fairness
Fairness <- cces %>% select(c("treatf", "justice"))
psych::alpha(Fairness)
```

```
## 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 = Fairness)
##
##   raw_alpha std.alpha G6(smc) average_r  S/N   ase mean   sd median_r
##       0.25      0.25    0.14      0.14 0.33 0.055  3.2 0.93     0.14
##
##  lower alpha upper     95% confidence boundaries
## 0.14 0.25 0.35
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## treatf       0.14      0.14    0.02      0.14  NA       NA  0.14  0.14
## justice      0.02      0.14      NA        NA  NA       NA  0.02  0.14
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean   sd
## treatf  130  0.88  0.76  0.28   0.14  3.2 1.06
## justice 131  0.85  0.76  0.28   0.14  3.2 0.95
##
```

```
## Non missing response frequency for each item
##            -1    0    1    2    3    4 miss
## treatf  0.02 0.02 0.03 0.12 0.28 0.53 0.82
## justice 0.01 0.01 0.05 0.12 0.39 0.43 0.82
```

```r
# Ingroup
Ingroup <- cces %>% select(c("proudc", "loyalf"))
psych::alpha(Ingroup)
```

```
## 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 = Ingroup)
##
##   raw_alpha std.alpha G6(smc) average_r  S/N   ase mean  sd median_r
##       0.26      0.27    0.15      0.15 0.36 0.053  2.4 1.3     0.15
##
##  lower alpha upper     95% confidence boundaries
## 0.16 0.26 0.36
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## proudc     0.154      0.15   0.024      0.15  NA       NA 0.154  0.15
## loyalf     0.024      0.15      NA        NA  NA       NA 0.024  0.15
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## proudc 147  0.79  0.76   0.3   0.15  3.0 1.2
## loyalf 129  0.86  0.76   0.3   0.15  1.8 1.6
##
## Non missing response frequency for each item
##           -1    0    1    2    3    4 miss
## proudc 0.02 0.03 0.08 0.16 0.26 0.46 0.80
## loyalf 0.10 0.12 0.21 0.18 0.20 0.19 0.83
```

```r
# Authority
Authority <- cces %>% select(c("authc", "roles"))
psych::alpha(Authority)
```

```
## 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 = Authority)
```

```
##
##   raw_alpha std.alpha G6(smc) average_r  S/N  ase mean  sd median_r
##       0.22      0.25    0.14      0.14 0.33 0.05  2.5 1.4     0.14
##
##  lower alpha upper     95% confidence boundaries
## 0.12 0.22 0.32
##
##  Reliability if an item is dropped:
##       raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## authc      0.14      0.14    0.02      0.14  NA       NA  0.14  0.14
## roles      0.02      0.14      NA        NA  NA       NA  0.02  0.14
##
##  Item statistics
##         n raw.r std.r r.cor r.drop mean  sd
## authc 127  0.67  0.76  0.28   0.14  3.3 1.1
## roles 135  0.87  0.76  0.28   0.14  1.7 1.8
##
## Non missing response frequency for each item
##         -1    0    1    2    3    4 miss
## authc 0.01 0.02 0.03 0.13 0.19 0.61 0.83
## roles 0.19 0.12 0.08 0.22 0.19 0.20 0.82
```

```r
# Purity
Purity <- cces %>% select(c("disgust", "notnat"))
psych::alpha(Purity)
```

```
## 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 = Purity)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.54      0.54    0.37      0.37 1.2 0.034  2.8 1.6     0.37
##
##  lower alpha upper     95% confidence boundaries
## 0.47 0.54 0.6
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## disgust      0.37      0.37    0.14      0.37  NA       NA  0.37  0.37
## notnat       0.14      0.37      NA        NA  NA       NA  0.14  0.37
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
```

```
## disgust 124  0.81  0.83   0.5   0.37  3.8 1.5
## notnat  118  0.83  0.83   0.5   0.37  1.7 1.7
##
## Non missing response frequency for each item
##           -1   0    1    2    3    4    5    6 miss
## disgust 0.00 0.0 0.10 0.13 0.22 0.19 0.23 0.15 0.83
## notnat  0.17 0.1 0.14 0.25 0.17 0.18 0.00 0.00 0.84
```

Repeated Measures GLM

```
# Individualizing and Binding scores
cces$indiv2 <- rowMeans(cces[, c("compat", "hurt", "treatf",
    "justice")], na.rm = TRUE)
cces$bind2 <- rowMeans(cces[, c("proudc", "loyalf", "authc",
    "roles", "disgust", "notnat")], na.rm = TRUE)

# Create a difference score
cces$diffscore2 <- cces$indiv2 - cces$bind2

# Run the linear model
diff.model2 <- lm(diffscore2 ~ ideo5, data = cces)
summary(diff.model2)
```

```
##
## Call:
## lm(formula = diffscore2 ~ ideo5, data = cces)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1041 -0.7175 -0.0741  0.8658  3.9259
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.01401    0.26610   7.569 1.37e-12 ***
## ideo5       -0.46996    0.07892  -5.955 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.348 on 199 degrees of freedom
##   (538 observations deleted due to missingness)
## Multiple R-squared:  0.1512, Adjusted R-squared:  0.147
## F-statistic: 35.46 on 1 and 199 DF,  p-value: 1.161e-08
```

```r
etaSquared(diff.model2)
```

```
##          eta.sq eta.sq.part
## ideo5 0.1512391   0.1512391
```

The procedure here is largely the same as the one in the 30-item section. The reported results are as follows:

- Aggregate difference between Indivdualizing and binding foundation: $F(1, 199) = 65.84$, p $<$ .001
- Moderation by Politics: $F(1, 199) = 34.56$, p $<$ .001, $\eta^2 = .15$