Moral Relevance: Duke CCES 2012

Jennifer Lin

11/25/2019

Contents

Introduction

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

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

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

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

```
library(car)
library(dplyr)
library(psych)
library(ggplot2)
library(GGally)
library("ggpubr")
library("reshape2")
library(scales)
library(lme4)  #lmer function
library(lsr)  #EtaSquared
library(coefplot)
```

Moral Relevance – 30 item

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

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

table(cces$math)
```

```
##
##   1   2   3   4   5   6
## 198 101  73  49  20   7
```

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

table(cces$dogood)
```

```
##
##   1   3   4   5   6
##   2   3  10  32 177
```

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

```
################ Recode Moral Relevance Items #####################

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

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

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

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

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

Previously, I recoded the scores on each foundation and created overall foundation scores. This will come in handy for the graph of descriptive statistics. But while we are at the recoding stage, I will create a score of the individualizing and binding foundations here.

```r
# Create Individualizing (indiv) and Binding (bind)
# foundation scores
cces$indiv <- rowMeans(cces[, c("emote", "weak", "cruel", "treatd",
    "unfair", "rights")], na.rm = TRUE)
cces$bind <- rowMeans(cces[, c("lovec", "betray", "loyal", "auth",
    "conform", "chaos", "pure", "disgust", "goddis")], na.rm = TRUE)
```

Cronbach's Alpha Calculations

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

```r
# Harm
Harm <- cces %>% select(c("emote", "weak", "cruel"))
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
##       0.72      0.72    0.64      0.46 2.6 0.018  3.2 1.2      0.5
##
##  lower alpha upper     95% confidence boundaries
## 0.68 0.72 0.75
##
##  Reliability if an item is dropped:
```

```
##       raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## emote      0.66      0.67    0.50      0.50 2.0    0.024    NA  0.50
## weak       0.52      0.53    0.36      0.36 1.1    0.034    NA  0.36
## cruel      0.68      0.68    0.52      0.52 2.1    0.024    NA  0.52
##
##  Item statistics
##         n raw.r std.r r.cor r.drop mean  sd
## emote 127  0.82  0.78  0.60   0.52  2.7 1.5
## weak  134  0.88  0.84  0.73   0.61  2.9 1.4
## cruel 132  0.78  0.78  0.59   0.50  3.9 1.2
##
## Non missing response frequency for each item
##          0    1    2    3    4    5 miss
## emote 0.12 0.09 0.18 0.24 0.27 0.09 0.83
## weak  0.08 0.10 0.17 0.24 0.28 0.12 0.82
## cruel 0.03 0.00 0.12 0.11 0.41 0.33 0.82
```

```r
# Fairness
Fairness <- cces %>% select(c("treatd", "unfair", "rights"))
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
##        0.7       0.7    0.61      0.44 2.4 0.019  3.5 1.3     0.43
##
##  lower alpha upper     95% confidence boundaries
## 0.67 0.7 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      0.58      0.58    0.41      0.41 1.4    0.031    NA  0.41
## unfair      0.60      0.60    0.43      0.43 1.5    0.029    NA  0.43
## rights      0.66      0.66    0.49      0.49 1.9    0.025    NA  0.49
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## treatd 126  0.88  0.81  0.65   0.55  3.4 1.4
## unfair 116  0.90  0.80  0.63   0.53  3.5 1.3
## rights 134  0.87  0.77  0.58   0.49  3.8 1.2
##
## Non missing response frequency for each item
##           0    1    2    3    4    5 miss
```

```
## treatd 0.06 0.04 0.11 0.25 0.34 0.20 0.83
## unfair 0.06 0.02 0.14 0.17 0.41 0.21 0.84
## rights 0.01 0.04 0.12 0.12 0.37 0.34 0.82
```

```r
# Ingroup
Ingroup <- cces %>% select(c("lovec", "betray", "loyal"))
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
##       0.92      0.92    0.92       0.8  12 0.0052  2.8 1.4     0.79
##
##  lower alpha upper     95% confidence boundaries
## 0.91 0.92 0.93
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## lovec       0.96      0.96    0.92      0.92 24.5   0.0029    NA  0.92
## betray      0.88      0.88    0.79      0.79  7.7   0.0085    NA  0.79
## loyal       0.82      0.82    0.69      0.69  4.5   0.0133    NA  0.69
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## lovec  132  0.91  0.89  0.79   0.76  2.6 1.5
## betray 138  0.93  0.94  0.92   0.85  3.1 1.5
## loyal  134  0.94  0.97  0.97   0.93  2.6 1.5
##
## Non missing response frequency for each item
##           0    1    2    3    4    5 miss
## lovec  0.14 0.10 0.21 0.24 0.20 0.10 0.82
## betray 0.09 0.06 0.20 0.20 0.25 0.20 0.81
## loyal  0.13 0.10 0.18 0.28 0.22 0.09 0.82
```

```r
# Authority
Authority <- cces %>% select(c("auth", "conform", "chaos"))
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
##       0.69      0.69    0.61      0.42 2.2 0.02  2.7 1.3     0.46
```

```
##
##  lower alpha upper     95% confidence boundaries
## 0.65 0.69 0.73
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## auth         0.63      0.63    0.46      0.46 1.73    0.027    NA  0.46
## conform      0.67      0.67    0.51      0.51 2.05    0.024    NA  0.51
## chaos        0.46      0.47    0.30      0.30 0.88    0.039    NA  0.30
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## auth    130  0.84  0.77  0.58   0.47  2.8 1.5
## conform 131  0.82  0.75  0.54   0.44  1.9 1.4
## chaos   128  0.84  0.84  0.72   0.60  3.3 1.4
##
## Non missing response frequency for each item
##            0    1    2    3    4    5 miss
## auth    0.11 0.10 0.15 0.26 0.25 0.13 0.82
## conform 0.17 0.22 0.31 0.15 0.13 0.03 0.82
## chaos   0.06 0.03 0.17 0.21 0.34 0.18 0.83
```

```r
# Purity
Purity <- cces %>% select(c("pure", "disgust", "goddis"))
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
##       0.58      0.58    0.49      0.32 1.4 0.027  2.7 1.4     0.35
##
##  lower alpha upper     95% confidence boundaries
## 0.52 0.58 0.63
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
## pure         0.39      0.40    0.25      0.25 0.67    0.044    NA  0.25
## disgust      0.51      0.52    0.35      0.35 1.09    0.035    NA  0.35
## goddis       0.52      0.52    0.35      0.35 1.10    0.035    NA  0.35
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## pure    132  0.80  0.77  0.58   0.45  3.2 1.5
```

```
## disgust 124  0.77  0.72  0.48   0.36  2.8 1.5
## goddis  124  0.83  0.72  0.48   0.37  2.1 1.9
##
## Non missing response frequency for each item
##            0    1    2    3    4    5 miss
## pure    0.06 0.11 0.14 0.17 0.26 0.25 0.82
## disgust 0.10 0.13 0.22 0.19 0.23 0.15 0.83
## goddis  0.31 0.12 0.13 0.15 0.13 0.16 0.83
```

Linegraph – Descriptive Statisitcs

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

```
######################## Clean Ideology Variable ################################

table(cces$ideo5)
```

```
##
##  1  2  3  4  5
## 22 40 59 55 30
```

```
# 1 = Very Liberal, 5 = very conservative
str(cces$ideo5)
```

```
##  int [1:739] NA NA NA NA NA NA NA NA 3 NA ...
```

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

```
##  chr [1:739] NA NA NA NA NA NA NA NA "3" NA "4" NA "3" "4" NA NA NA ...
```

```
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")

table(cces$ideology)
```

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

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

```
##
##      Conservative           Liberal          Moderate Very Conservative
##                55                40                59                30
##      Very Liberal                NA
##                22               533
```

```r
# 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"))
```

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

```r
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)
```

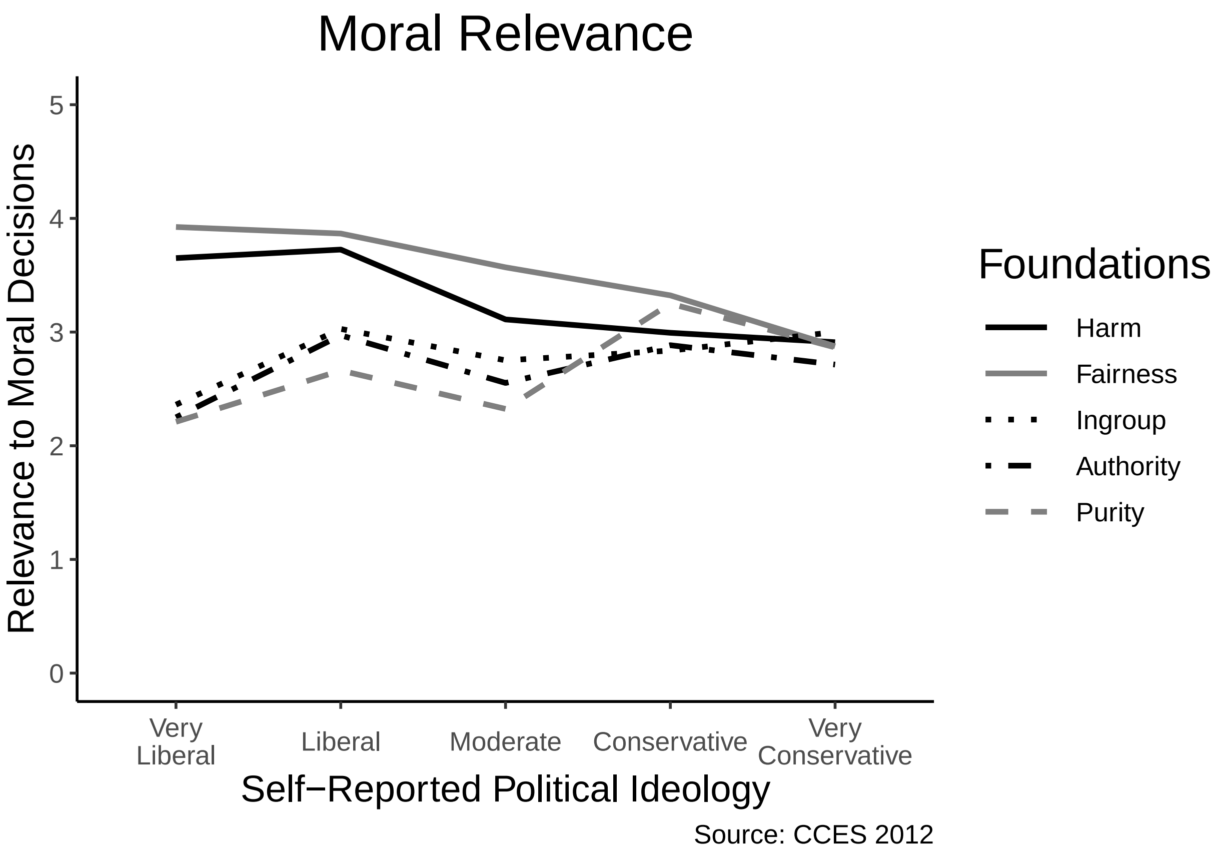
The above method creates 5 different data frames. To create a graph, we need everything to be on one data frame. Here, I will merge each of these data frames into one and reshape them into a format that `ggplot` will be able to read.

```r
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!

```r
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: CCES 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

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

```r
# 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
## -3.4553 -0.6434 -0.0434  0.6670  3.3329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.87899    0.20308   9.252  < 2e-16 ***
## ideo5       -0.41187    0.06019  -6.843 8.97e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.04 on 203 degrees of freedom
##   (534 observations deleted due to missingness)
## Multiple R-squared:  0.1874, Adjusted R-squared:  0.1834
## F-statistic: 46.83 on 1 and 203 DF,  p-value: 8.97e-11
```

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

```
##          eta.sq eta.sq.part
## ideo5 0.1874404   0.1874404
```

The results here are interpreted as follows:

The `F-statistic: 46.83 on 1 and 203 DF,  p-value: 8.97e-11` reflects the moderation by politics model.

To find the difference between the aggregate individualizing versus binding foundations, we square the t-value under `(Intercept)` and report the respective p-value.

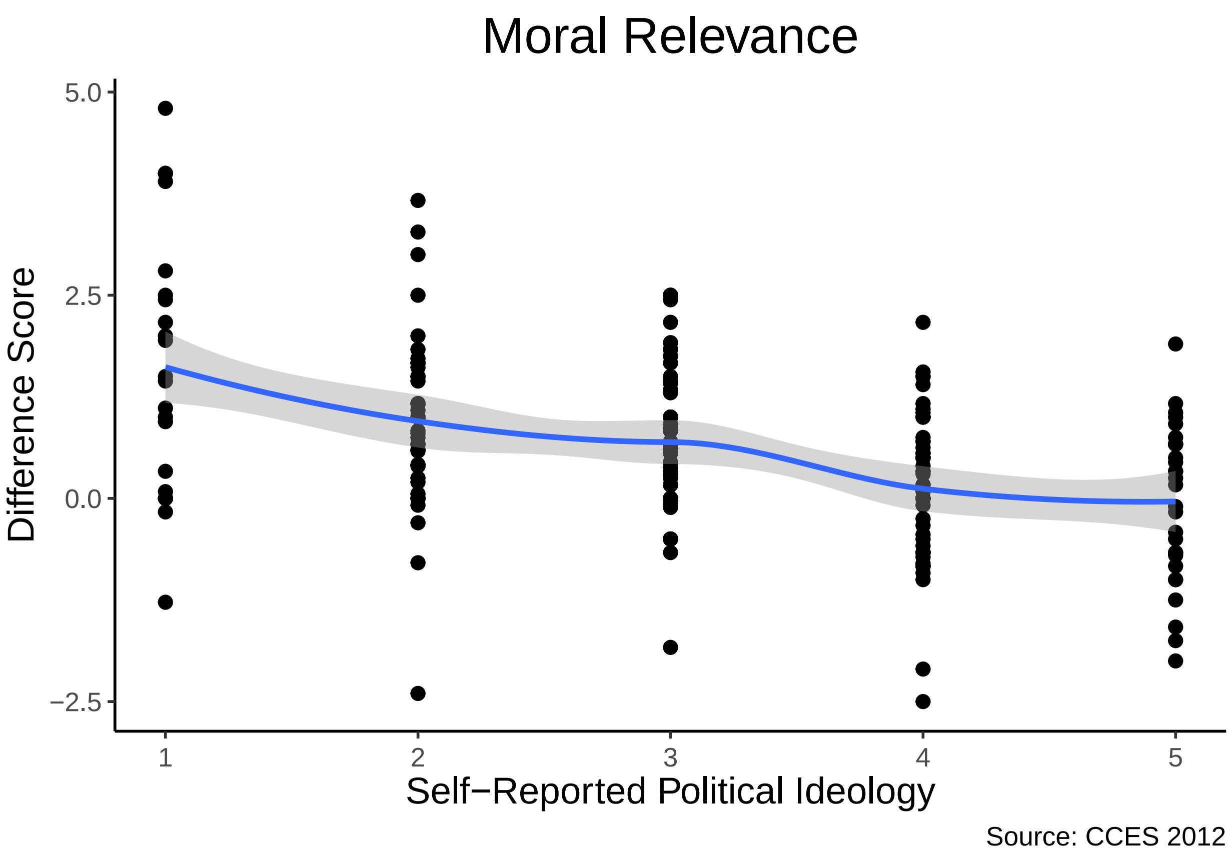
The results are: - Aggregate difference between individualizing and binding foundations: F(1, 203) = 85.599, p < .001 - Moderation by politics: F(1, 203) = 46.83, p < .001, η^2 = .187

With these results, we see that liberals and conservatives differ form one another in their moral foundations such that the more conservative one becomes, the less likely there are to be differences in their scores between the foundations.

To check our results, I create two graphs below that display the data with a loess (graph 1) and a best fitted (graph 2) line.

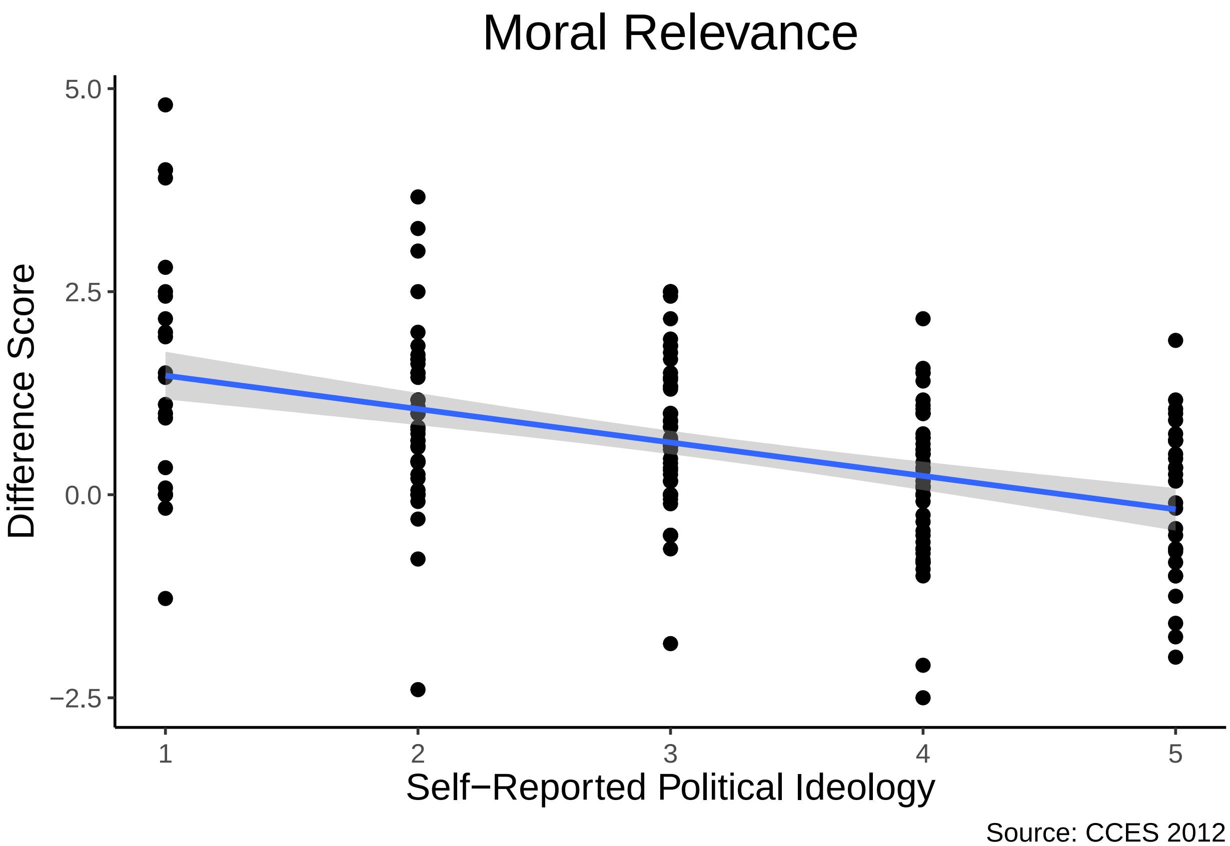
For the x-axis, higher values indicate a more conservative group such that 1 = very liberal and 5 = very conservative

```
# Fit plot with loess line
ggplot(cces, aes(x = ideo5, y = diffscore)) + geom_point(size = 2) +
    geom_smooth(method = "auto", se = TRUE, fullrange = FALSE,
        level = 0.95) + theme_classic() + ggtitle("Moral Relevance") +
    xlab("Self-Reported Political Ideology") + ylab("Difference Score") +
    labs(caption = "Source: CCES 2012") + 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"))
```




```
# Fit plot with linear regression line
ggplot(cces, aes(x = ideo5, 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: CCES 2012") + 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"))
```



Moral Relevance – 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.

```
################ Moral Relevance Items ######################

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

### Fairness ###
cces$treatd <- cces$treatd - 1
```

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

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

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

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

```r
# Create Individualizing (indiv) and Binding (bind)
# foundation scores for the 20-item
cces$indiv2 <- rowMeans(cces[, c("emote", "weak", "treatd", "unfair")],
    na.rm = TRUE)
cces$bind2 <- rowMeans(cces[, c("lovec", "betray", "auth", "conform",
    "pure", "disgust")], na.rm = TRUE)
```

Cronbach's Alpha Calculation

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

```r
# Harm
Harm2 <- cces %>% 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 0.024  1.8 1.3     0.52
##
##  lower alpha upper     95% confidence boundaries
```

```
## 0.63 0.68 0.73
##
##  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 127   0.9  0.87  0.62   0.52  1.7 1.5
## weak  134   0.9  0.87  0.62   0.52  1.9 1.4
##
## Non missing response frequency for each item
##         -1    0    1    2    3    4 miss
## emote 0.12 0.09 0.18 0.24 0.27 0.09 0.83
## weak  0.08 0.10 0.17 0.24 0.28 0.12 0.82
```

```
# Fairness
Fairness2 <- cces %>% 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
##       0.66      0.66    0.49      0.49 1.9 0.025  2.4 1.3     0.49
##
##  lower alpha upper     95% confidence boundaries
## 0.61 0.66 0.71
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## treatd      0.49      0.49    0.24      0.49  NA       NA  0.49  0.49
## unfair      0.24      0.49      NA        NA  NA       NA  0.24  0.49
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## treatd 126  0.94  0.86   0.6   0.49  2.4 1.4
## unfair 116  0.93  0.86   0.6   0.49  2.5 1.3
##
## Non missing response frequency for each item
##          -1    0    1    2    3    4 miss
```

```
## treatd 0.06 0.04 0.11 0.25 0.34 0.20 0.83
## unfair 0.06 0.02 0.14 0.17 0.41 0.21 0.84
```

```r
# Ingroup
Ingroup2 <- cces %>% 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
##       0.82      0.82    0.69      0.69 4.5 0.013  1.8 1.5     0.69
##
##  lower alpha upper     95% confidence boundaries
## 0.79 0.82 0.84
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## lovec       0.69      0.69    0.48      0.69  NA       NA  0.69  0.69
## betray      0.48      0.69      NA        NA  NA       NA  0.48  0.69
##
##  Item statistics
##          n raw.r std.r r.cor r.drop mean  sd
## lovec  132  0.94  0.92  0.77   0.69  1.6 1.5
## betray 138  0.94  0.92  0.77   0.69  2.1 1.5
##
## Non missing response frequency for each item
##          -1    0    1    2    3   4 miss
## lovec  0.14 0.10 0.21 0.24 0.20 0.1 0.82
## betray 0.09 0.06 0.20 0.20 0.25 0.2 0.81
```

```r
# Authority
Authority2 <- cces %>% 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
```

```
##       0.46      0.47     0.3       0.3 0.88 0.039  1.4 1.4      0.3
##
##  lower alpha upper     95% confidence boundaries
## 0.39 0.46 0.54
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## auth        0.305       0.3   0.093       0.3  NA       NA 0.305   0.3
## conform     0.093       0.3      NA        NA  NA       NA 0.093   0.3
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## auth    130  0.89  0.81  0.45    0.3 1.83 1.5
## conform 131  0.87  0.81  0.45    0.3 0.94 1.4
##
## Non missing response frequency for each item
##           -1    0    1    2    3    4 miss
## auth    0.11 0.10 0.15 0.26 0.25 0.13 0.82
## conform 0.17 0.22 0.31 0.15 0.13 0.03 0.82
```

```
# Purity
Purity2 <- cces %>% select(c("pure", "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
##       0.52      0.52    0.35      0.35 1.1 0.035  2.5 1.5     0.35
##
##  lower alpha upper     95% confidence boundaries
## 0.45 0.52 0.59
##
##  Reliability if an item is dropped:
##         raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## pure         0.35      0.35    0.13      0.35  NA       NA  0.35  0.35
## disgust      0.13      0.35      NA        NA  NA       NA  0.13  0.35
##
##  Item statistics
##           n raw.r std.r r.cor r.drop mean  sd
## pure    132  0.85  0.82  0.49   0.35  3.2 1.5
## disgust 124  0.85  0.82  0.49   0.35  1.8 1.5
```

```
##
## Non missing response frequency for each item
##           -1    0    1    2    3    4    5 miss
## pure    0.0 0.06 0.11 0.14 0.17 0.26 0.25 0.82
## disgust 0.1 0.13 0.22 0.19 0.23 0.15 0.00 0.83
```

Linegraph – Descriptive Statistics

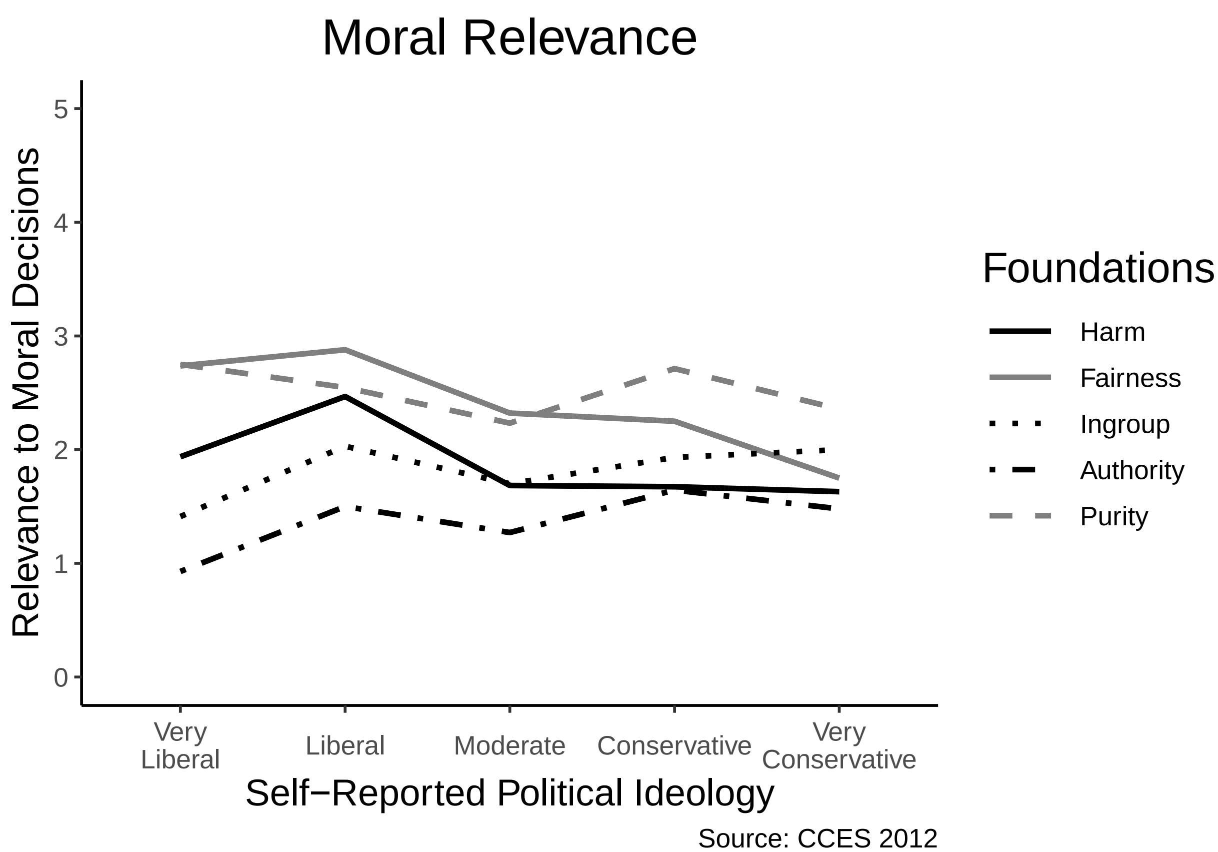
For the linegraph, I will use the same ideology variable created earlier. I begin by aggregating the results for the 20 item MFQ and reshaping the data.

```
############# 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")
```

Then, 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: CCES 201
```

```r
    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

I create a repeated measures GLM analysis using the same framework as above.

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

```r
# 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
## -3.3239 -0.7405  0.0095  0.7903  3.2595
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.11732    0.24398   4.580 8.36e-06 ***
## ideo5       -0.29226    0.07245  -4.034 7.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.217 on 192 degrees of freedom
##   (545 observations deleted due to missingness)
## Multiple R-squared:  0.07813,    Adjusted R-squared:  0.07333
## F-statistic: 16.27 on 1 and 192 DF,  p-value: 7.912e-05
```

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

```
##           eta.sq eta.sq.part
## ideo5 0.07813024  0.07813024
```

The results here are interpreted as follows:

The `F-statistic: 16.27 on 1 and 192 DF,  p-value: 7.912e-05` reflects the moderation by politics model.

To find the difference between the aggregate individualizing versus binding foundations, we square the t-value under `(Intercept)` and report the respective p-value.

The results are: - Aggregate difference between individualizing and binding foundations: F(1, 192) = 20.976, p < .001 - Moderation by politics: F(1, 192) = 16.27, p < .001, η^2 = .078

With these results, we see that liberals and conservatives differ form one another in their moral foundations such that the more conservative one becomes, the less likely there are to be differences in their scores between the foundations.

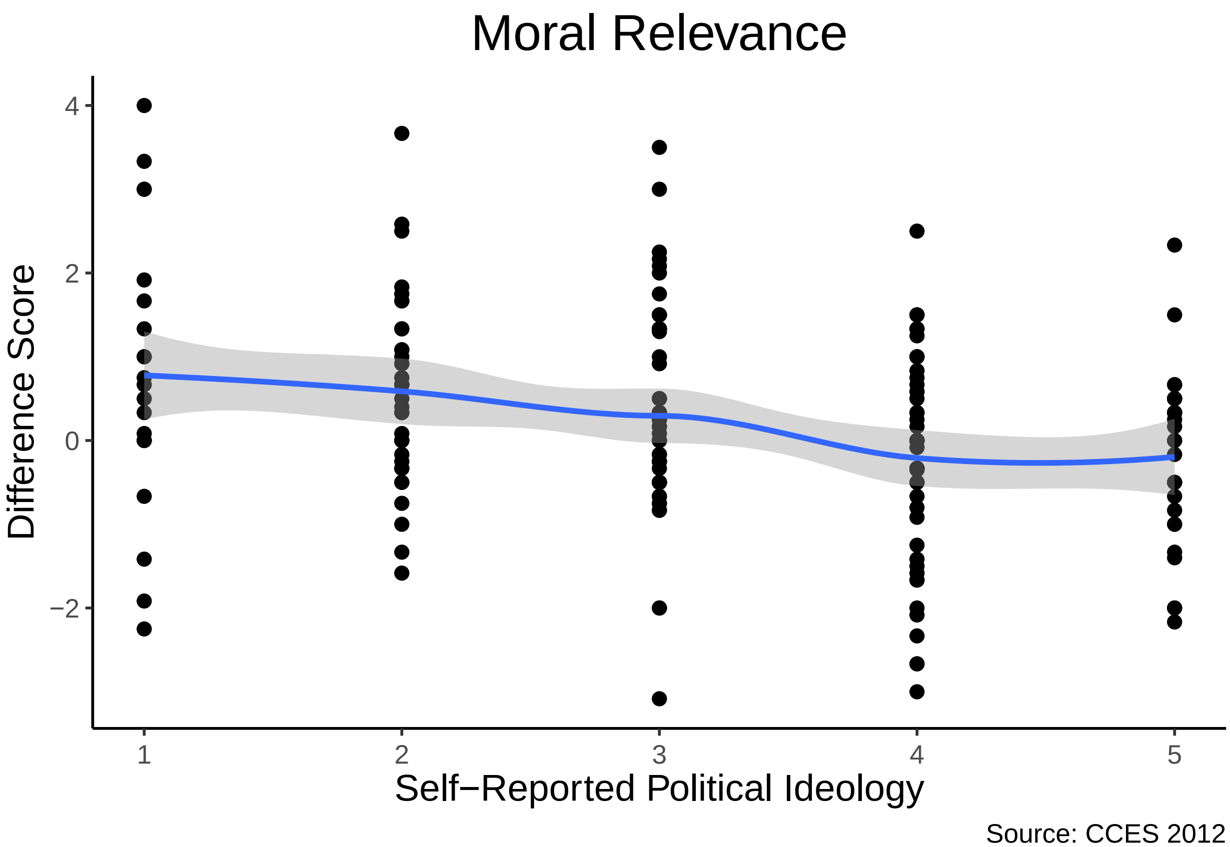
While the effect is smaller than the 30-item, there are differences that can be detected between the scale and between liberals and conservatives

To check our results, I create two graphs below that display the data with a loess (graph 1) and a best fitted (graph 2) line.

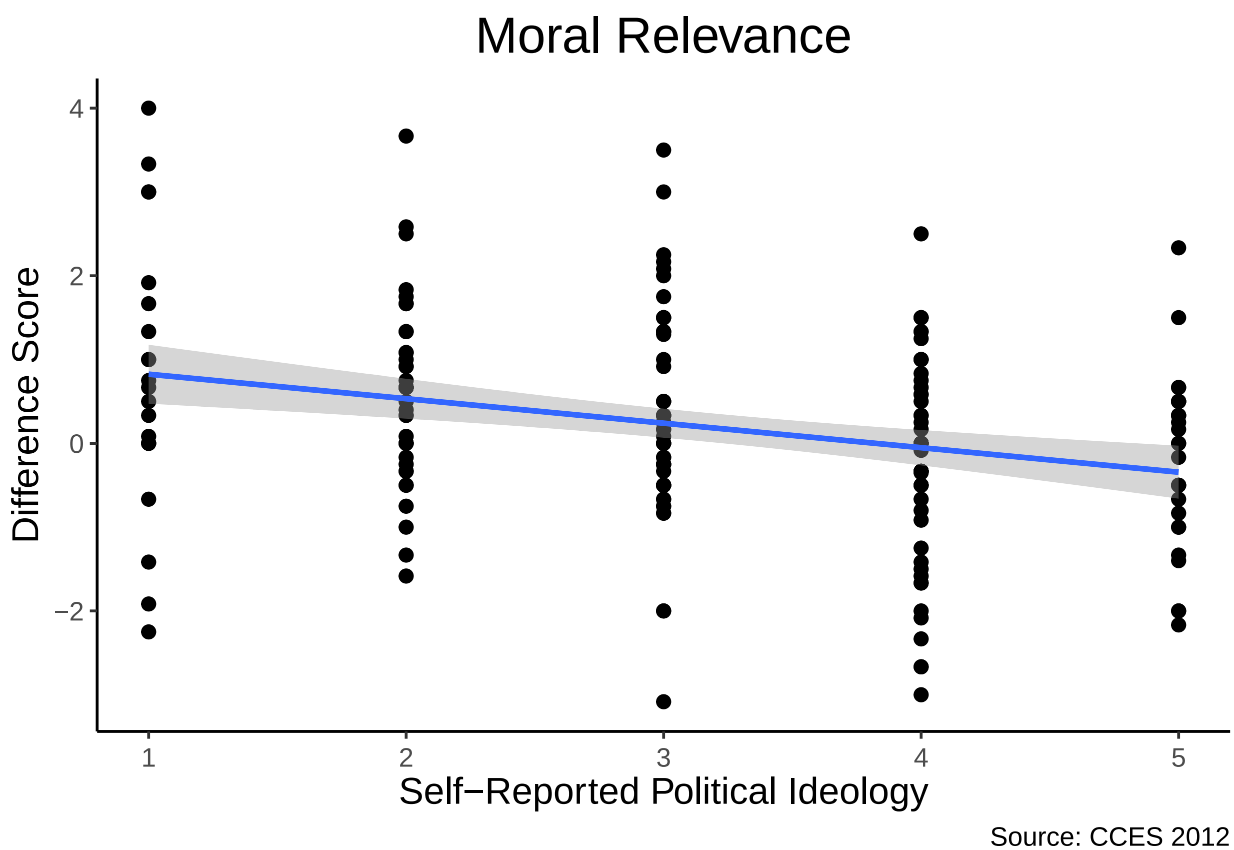
For the x-axis, higher values indicate a more conservative group such that 1 = very liberal and 5 = very conservative

```r
# Fit plot with loess line
ggplot(cces, aes(x = ideo5, y = diffscore2)) + geom_point(size = 2) +
    geom_smooth(method = "auto", se = TRUE, fullrange = FALSE,
        level = 0.95) + theme_classic() + ggtitle("Moral Relevance") +
    xlab("Self-Reported Political Ideology") + ylab("Difference Score") +
```

```r
    labs(caption = "Source: CCES 2012") + 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"))
```




```r
# Fit plot with linear regression line
ggplot(cces, aes(x = ideo5, y = diffscore2)) + 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: CCES 2012") + 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"))
```


Moral Relevance
4
2
0
−2
Difference Score
1
2
3
4
5
Self−Reported Political Ideology
Source: CCES 2012