# Tutorial 13: Simulations and Regression Discontinuity

*Jan Vogler (jan.vogler@duke.edu)*

*November 6, 2015*

## Today's Agenda

1. Model simulations
2. Using model simulations to make predictions
3. Estimating interaction effects through simulations
4. Regression discontinuity designs

**Credit to Professor Chris Johnston. The code for the simulation approach introduced here is from his class.**

## 1. Model simulations

We will look at data from a data set that we have not used previously, data on attitudes toward the United States Supreme Court.

```
setwd("C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/w13/")
courtdata <- read.table("courtdata.txt", header = TRUE)
summary(courtdata)
```

```
##      income           ideo           ruling              age
##  Min.   : 1.00   Min.   :1.000   Min.   :0.0000   Min.   :19.00
##  1st Qu.: 6.00   1st Qu.:3.000   1st Qu.:1.0000   1st Qu.:39.00
##  Median : 8.00   Median :3.000   Median :1.0000   Median :50.00
##  Mean   : 8.46   Mean   :3.128   Mean   :0.8249   Mean   :49.82
##  3rd Qu.:11.00   3rd Qu.:4.000   3rd Qu.:1.0000   3rd Qu.:59.00
##  Max.   :15.00   Max.   :5.000   Max.   :1.0000   Max.   :86.00
##       male             black             hisp              soph
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.: 4.000
##  Median :1.0000   Median :0.0000   Median :0.00000   Median : 7.000
##  Mean   :0.5065   Mean   :0.1065   Mean   :0.08284   Mean   : 6.724
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.: 9.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :10.000
##      sclaw          college
##  Min.   :1.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.0000
##  Median :2.000   Median :0.0000
##  Mean   :2.357   Mean   :0.3302
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :5.000   Max.   :1.0000
```

What is the meaning of the variables in the data set?

## Independent Variables

college = dummy for college education

ideo = 5-point ideological self-identification, ranging from “very liberal” to “very conservative”

soph = 10-item political knowledge scale

black, hispanic = dummies for respective categories

income = 15-point household income category scale

ruling = dummy indicating respondent correctly identified the Affordable Care Act ruling

male = dummy indicating male gender

## Dependent Variable

We are interested in attitudes of US citizens toward the Supreme Court. For this purpose we look at the variable “sclaw” (“Supreme Court Law”). The variable was based on the following statement:

“The Supreme Court should be allowed to throw out any law it deems unconstitutional”

The variable ranges from “strongly agree” to “strongly disagree”.

Let us estimate a linear model.

```r
model1 <- lm(sclaw ~ ruling + ideo + soph + age + male + black + hisp + college +
    income, data = courtdata)
summary(model1)
```

```
##
## Call:
## lm(formula = sclaw ~ ruling + ideo + soph + age + male + black +
##     hisp + college + income, data = courtdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9065 -1.1406 -0.3534  0.8163  3.2797
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.919930   0.240242  12.154  < 2e-16 ***
## ruling      -0.133350   0.135592  -0.983  0.32567
## ideo        -0.135789   0.045475  -2.986  0.00291 **
## soph        -0.036333   0.020474  -1.775  0.07633 .
## age          0.004780   0.003167   1.509  0.13161
## male        -0.165054   0.100291  -1.646  0.10019
## black        0.071503   0.153606   0.465  0.64170
## hisp        -0.123167   0.170601  -0.722  0.47052
## college     -0.039149   0.105591  -0.371  0.71091
## income       0.009154   0.012308   0.744  0.45721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.34 on 835 degrees of freedom
## Multiple R-squared:  0.02962,    Adjusted R-squared:  0.01916
## F-statistic: 2.832 on 9 and 835 DF,  p-value: 0.002746
```

When we estimate a linear model, all our coefficients are assumed to be normally distributed random variables with a mean and a variance that depends on the data. We can make use of this fact by simulating different versions of the model.

For the simulations we need the package “arm”. Please make sure to install it via the following command: install.packages(“arm”)

```
library(arm)
```

```
## Warning: package 'arm' was built under R version 3.2.2
```

```
## Loading required package: MASS
## Loading required package: Matrix
## Loading required package: lme4
```

```
## Warning: package 'lme4' was built under R version 3.2.2
```

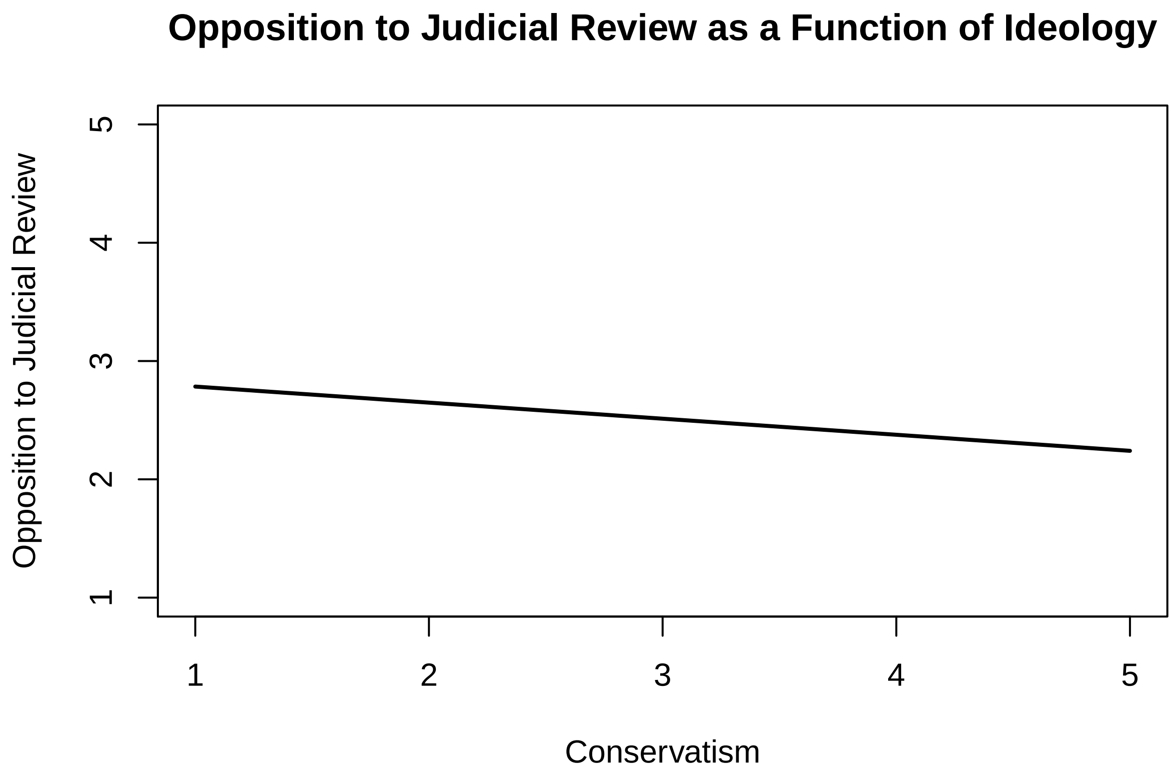
```
##
## arm (Version 1.8-6, built: 2015-7-7)
##
## Working directory is C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/W13
```

```
model1.sims <- sim(model1, n.sims = 1000)
```

## 2. Using model simulations to make predictions

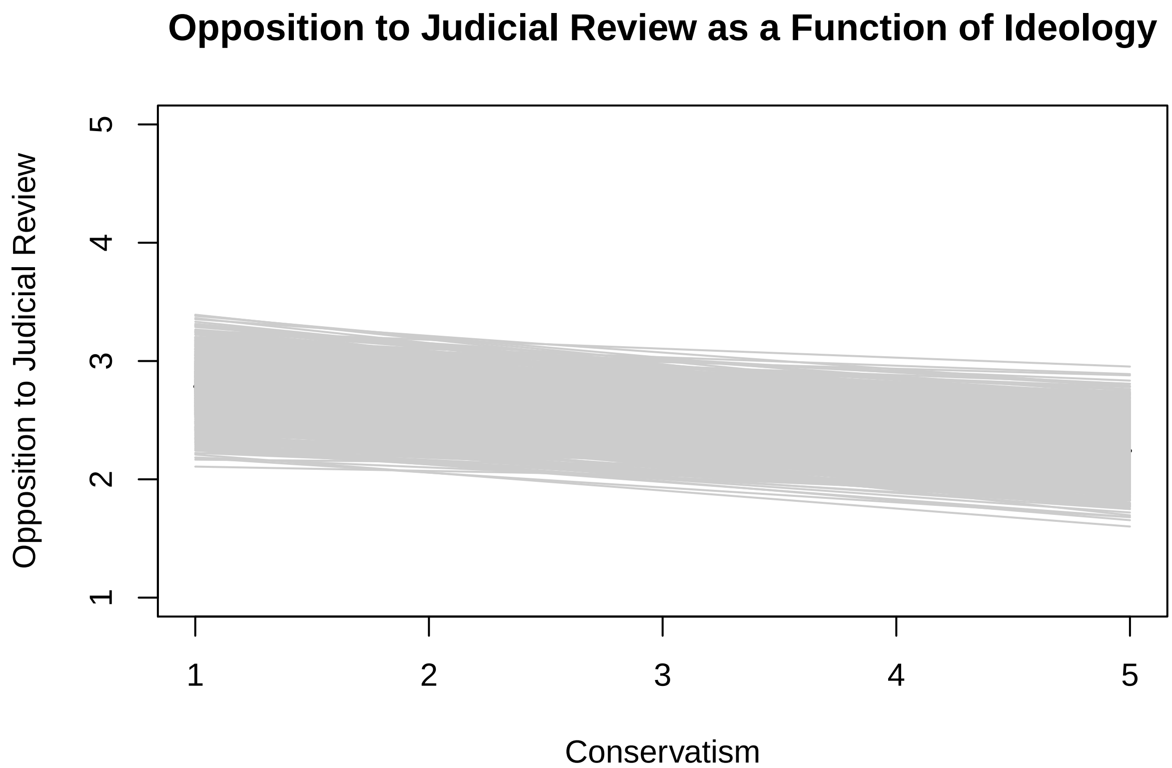
The following plot is generated by our knowledge about the regression. We access the first coefficient (the intercept) and the third coefficient (ideology). We let ideology vary from 1 to 5. If we plug in the right formula, then we will get the predicted values when all other variables are at the value 0.

```
curve(coef(model1)[1] + coef(model1)[3] * x, from = 1, to = 5, ylim = c(1, 5),
    xlab = "Conservatism", ylab = "Opposition to Judicial Review", main = "Opposition to Judicial Revie
    lwd = 2)
```




Now let's look instead at the lines that are the result of the 1000 simulations that we created above.

```
curve(coef(model1)[1] + coef(model1)[3] * x, from = 1, to = 5, ylim = c(1, 5),
    xlab = "Conservatism", ylab = "Opposition to Judicial Review", main = "Opposition to Judicial Revie
    lwd = 2)
for (i in 1:1000) {
    curve(coef(model1.sims)[i, 1] + coef(model1.sims)[i, 3] * x, add = TRUE,
        col = "gray80")
}
```



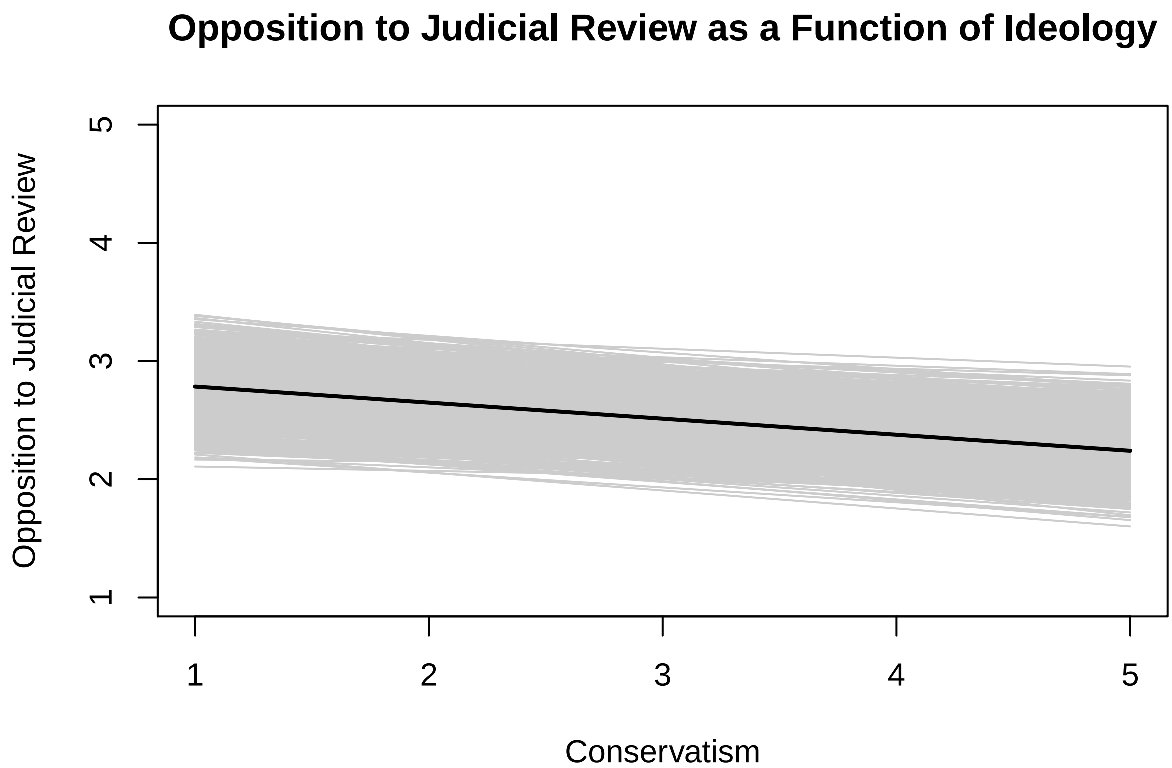

What we can see in this plot is that the simulated coefficients are very similar to the one that we already had. In fact, the simulation predictions are approximately normally distributed around our original regression line.

Let's run the original command one more time, so we get our regression line back.

```
curve(coef(model1)[1] + coef(model1)[3] * x, from = 1, to = 5, ylim = c(1, 5),
    xlab = "Conservatism", ylab = "Opposition to Judicial Review", main = "Opposition to Judicial Revie
    lwd = 2)

curve(coef(model1)[1] + coef(model1)[3] * x, from = 1, to = 5, ylim = c(1, 5),
    xlab = "Conservatism", ylab = "Opposition to Judicial Review", main = "Opposition to Judicial Revie
    lwd = 2)
for (i in 1:1000) {
    curve(coef(model1.sims)[i, 1] + coef(model1.sims)[i, 3] * x, add = TRUE,
        col = "gray80")
}

curve(coef(model1)[1] + coef(model1)[3] * x, col = "black", lwd = 2, add = TRUE)
```




Let us now take our model simulations into account. They can help us to make an average predictive comparison. We use our model simulations in combination with draws from the data to create a so-called "average predictive comparison".

```
# Using simulation to get CIs for average predictive comparisons Create a
# matrix and a vector to be filled with (1) 1000 simulations of change in
# predicted prob, 1 for each obs, and (2) 1000 simulations of AVP

# Remember, these were the variables from our regression: (intercept) +
# ruling + ideo + soph + age + male + black + hisp + college + income

summary(courtdata$ideo)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.128   4.000   5.000
```

```
d.ideo <- array(NA, c(1000, length(courtdata$sclaw)))
m.ideo <- array(NA, 1000)

for (i in 1:1000) {
    d.ideo[i, ] <- (coef(model1.sims)[i, 1] + coef(model1.sims)[i, 2] * courtdata$ruling +
        coef(model1.sims)[i, 3] * 5 + coef(model1.sims)[i, 4] * courtdata$soph +
        coef(model1.sims)[i, 5] * courtdata$age + coef(model1.sims)[i, 6] *
        courtdata$male + coef(model1.sims)[i, 7] * courtdata$black + coef(model1.sims)[i,
        8] * courtdata$hisp + coef(model1.sims)[i, 9] * courtdata$college +
        coef(model1.sims)[i, 10] * courtdata$income) - (coef(model1.sims)[i,
```

```r
        1] + coef(model1.sims)[i, 2] * courtdata$ruling + coef(model1.sims)[i,
        3] * 0 + coef(model1.sims)[i, 4] * courtdata$soph + coef(model1.sims)[i,
        5] * courtdata$age + coef(model1.sims)[i, 6] * courtdata$male + coef(model1.sims)[i,
        7] * courtdata$black + coef(model1.sims)[i, 8] * courtdata$hisp + coef(model1.sims)[i,
        9] * courtdata$college + coef(model1.sims)[i, 10] * courtdata$income)
    m.ideo[i] <- mean(d.ideo[i, ])
}

mean(m.ideo)
```

```
## [1] -0.6630109
```

```r
sd(m.ideo)
```

```
## [1] 0.2245443
```

```r
quantile(m.ideo, probs = c(0.025, 0.16, 0.84, 0.975))
```

```
##       2.5%        16%        84%      97.5%
## -1.1074764 -0.8861177 -0.4353768 -0.2354961
```

Let's compare this to the effect that sophistication has.

```r
summary(courtdata$soph)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000   7.000   6.724   9.000  10.000
```

```r
d.soph <- array(NA, c(1000, length(courtdata$sclaw)))
m.soph <- array(NA, 1000)

for (i in 1:1000) {
    d.soph[i, ] <- ((coef(model1.sims)[i, 1] + coef(model1.sims)[i, 2] * courtdata$ruling +
        coef(model1.sims)[i, 3] * courtdata$ideo + coef(model1.sims)[i, 4] *
        10 + coef(model1.sims)[i, 5] * courtdata$age + coef(model1.sims)[i,
        6] * courtdata$male + coef(model1.sims)[i, 7] * courtdata$black + coef(model1.sims)[i,
        8] * courtdata$hisp + coef(model1.sims)[i, 9] * courtdata$college +
        coef(model1.sims)[i, 10] * courtdata$income) - (coef(model1.sims)[i,
        1] + coef(model1.sims)[i, 2] * courtdata$ruling + coef(model1.sims)[i,
        3] * courtdata$ideo + coef(model1.sims)[i, 4] * 0 + coef(model1.sims)[i,
        5] * courtdata$age + coef(model1.sims)[i, 6] * courtdata$male + coef(model1.sims)[i,
        7] * courtdata$black + coef(model1.sims)[i, 8] * courtdata$hisp + coef(model1.sims)[i,
        9] * courtdata$college + coef(model1.sims)[i, 10] * courtdata$income))
    m.soph[i] <- mean(d.soph[i, ])
}

mean(m.soph)
```

```
## [1] -0.362254
```

```r
sd(m.soph)
```

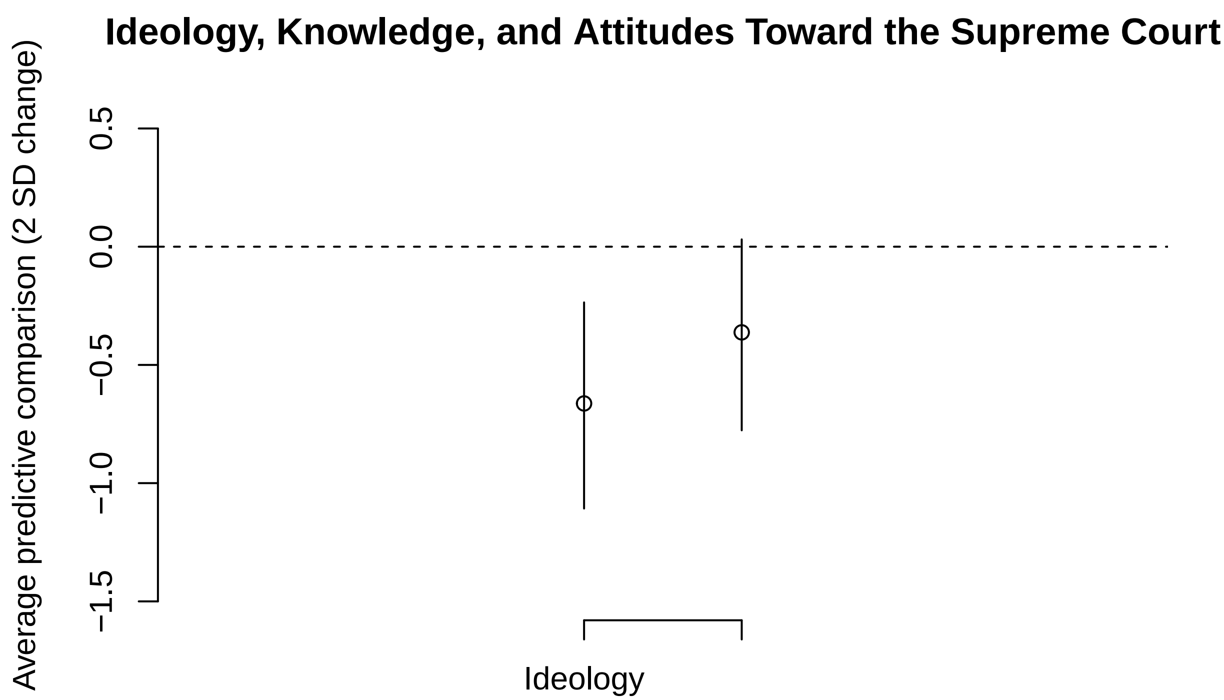
```
## [1] 0.2035271
```

```r
quantile(m.soph, probs = c(0.025, 0.16, 0.84, 0.975))
```

```
##        2.5%         16%         84%       97.5%
## -0.77669708 -0.55775067 -0.15510270  0.03102466
```

Let us plot these two in comparison.

```r
plot(1:2, c(mean(m.ideo), mean(m.soph)), type = "p", ylim = c(-1.5, 0.5), xlab = "",
    main = "Ideology, Knowledge, and Attitudes Toward the Supreme Court", ylab = "Average predictive co
    asp = 1.5, axes = FALSE)
axis(1, at = c(1, 2), labels = c("Ideology", "Knowledge"))
axis(2, at = c(-1.5, -1, -0.5, 0, 0.5))
abline(h = 0, lty = 2)
segments(1, quantile(m.ideo, probs = c(0.025)), 1, quantile(m.ideo, probs = c(0.975)))
segments(2, quantile(m.soph, probs = c(0.025)), 2, quantile(m.soph, probs = c(0.975)))
```




## 3. Estimating interaction effects through simulations

Let us consider an interaction effect of ideology and political sophistication.

```r
model2 <- lm(sclaw ~ ruling + ideo + soph + ideo:soph + age + male + black +
    hisp + college + income, data = courtdata)
summary(model2)
```

```
##
## Call:
## lm(formula = sclaw ~ ruling + ideo + soph + ideo:soph + age +
##     male + black + hisp + college + income, data = courtdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8656 -1.1373 -0.3692  0.8050  3.4347
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.234052   0.439457   5.084 4.57e-07 ***
## ruling      -0.129621   0.135407  -0.957   0.3387
## ideo         0.078416   0.123634   0.634   0.5261
## soph         0.059929   0.055574   1.078   0.2812
## age          0.005185   0.003170   1.636   0.1023
## male        -0.167828   0.100154  -1.676   0.0942 .
## black        0.078785   0.153429   0.513   0.6077
## hisp        -0.121305   0.170352  -0.712   0.4766
## college     -0.049719   0.105588  -0.471   0.6379
## income       0.007989   0.012306   0.649   0.5164
## ideo:soph   -0.030507   0.016377  -1.863   0.0628 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.338 on 834 degrees of freedom
## Multiple R-squared:  0.03364,    Adjusted R-squared:  0.02205
## F-statistic: 2.903 on 10 and 834 DF,  p-value: 0.001406
```

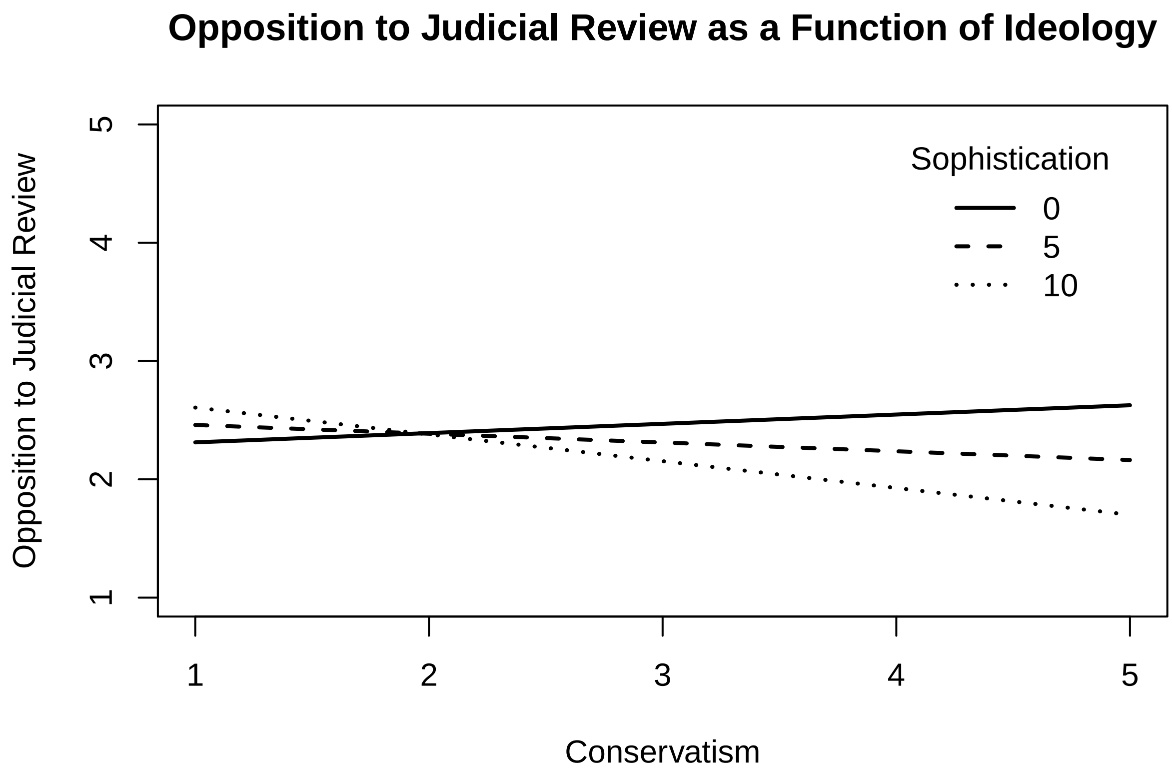
As we can see, there is some statistical evidence for an interaction effect.

Let's plot this interaction effect based on the regression output.

```r
curve(coef(model2)[1] + coef(model2)[3] * x + coef(model2)[4] * 0 + coef(model2)[11] *
    x * 0, from = 1, to = 5, ylim = c(1, 5), xlab = "Conservatism", ylab = "Opposition to Judicial Revi
    main = "Opposition to Judicial Review as a Function of Ideology", lwd = 2)

curve(coef(model2)[1] + coef(model2)[3] * x + coef(model2)[4] * 5 + coef(model2)[11] *
    x * 5, lty = 2, lwd = 2, add = TRUE)
curve(coef(model2)[1] + coef(model2)[3] * x + coef(model2)[4] * 10 + coef(model2)[11] *
    x * 10, lty = 3, lwd = 2, add = TRUE)

legend("topright", title = "Sophistication", c("0", "5", "10"), lty = c(1, 2,
    3), lwd = c(2, 2, 2), bty = "n", inset = 0.05)
```




What we can see in the picture is the effect that political ideology has on the opposition to judicial review at different levels of sophistication.

Lets use another approach in which we take mode simulations into account.

```
model2.sims <- sim(model2, n.sims = 1000)

# Create blank matrix for ideo beta sims

soph.sims <- matrix(NA, nrow = 1000, ncol = 11)

# Fill matrix, where each column corresponds with one of the 11 values of
# soph

for (i in 0:10) {
    soph.sims[, i + 1] <- coef(model2.sims)[, 3] + coef(model2.sims)[, 11] *
        i
}

# Summarize conditional coefficients for ideo

ideo.betas <- matrix(NA, nrow = 11, ncol = 1)
for (i in 1:11) {
    ideo.betas[i, 1] <- mean(soph.sims[, i])
}

ideo.ci <- matrix(NA, nrow = 11, ncol = 4)
```

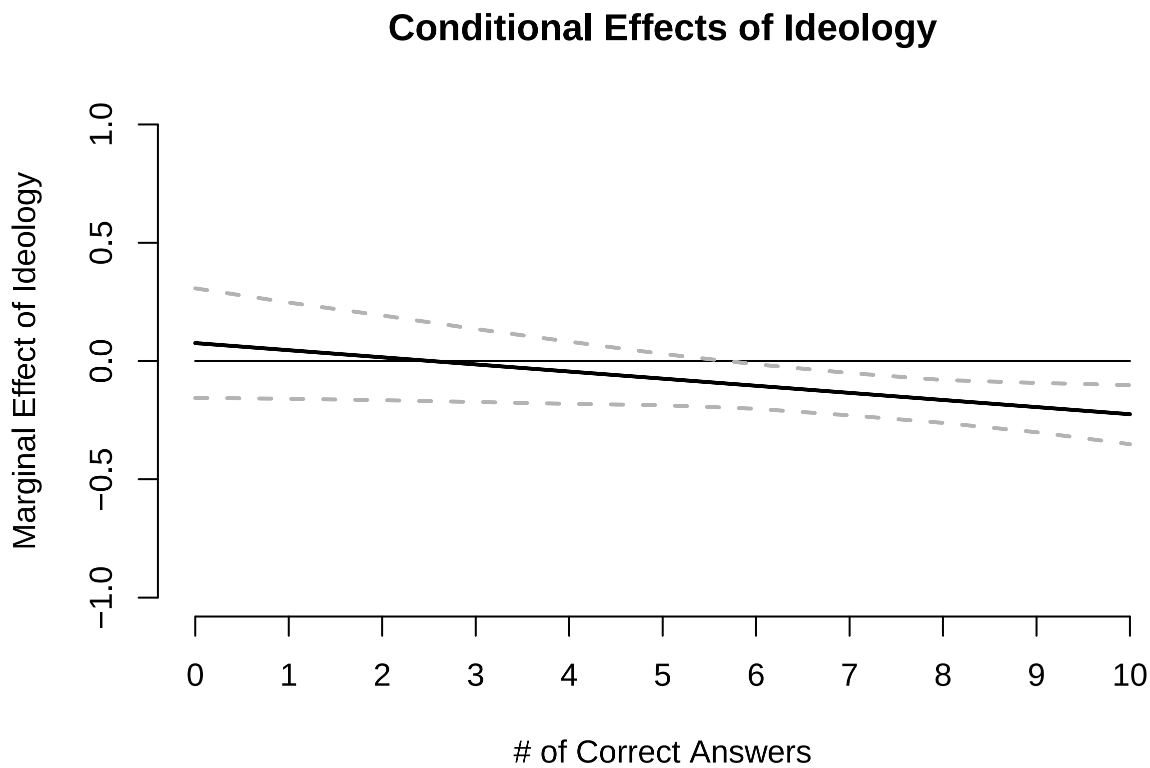
```r
for (i in 1:11) {
    ideo.ci[i, 1] <- quantile(soph.sims[, i], 0.025)
    ideo.ci[i, 2] <- quantile(soph.sims[, i], 0.975)
    ideo.ci[i, 3] <- quantile(soph.sims[, i], 0.16)
    ideo.ci[i, 4] <- quantile(soph.sims[, i], 0.84)
}

ideo.table <- cbind(ideo.betas, ideo.ci)
ideo.table
```

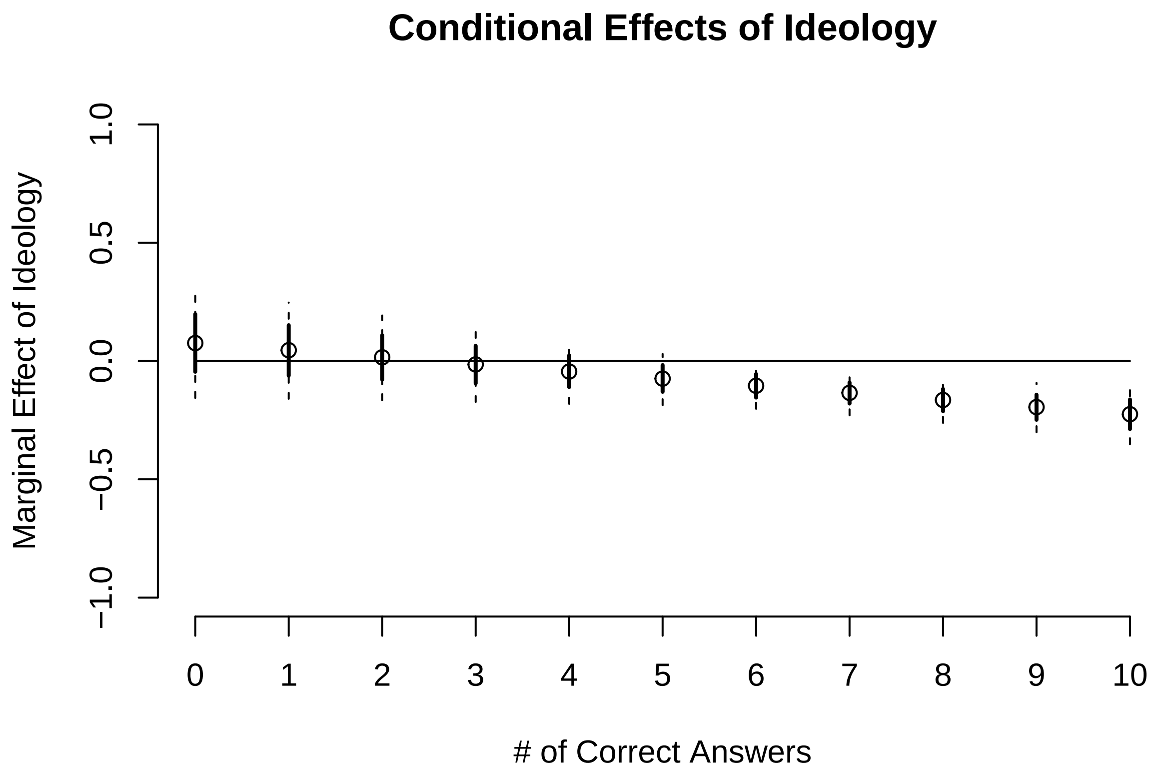
```
##              [,1]       [,2]        [,3]        [,4]        [,5]
##  [1,]  0.07584759 -0.1559076  0.30722273 -0.04533409  0.19819477
##  [2,]  0.04579058 -0.1595699  0.24710625 -0.06187775  0.15146620
##  [3,]  0.01573356 -0.1654340  0.19259451 -0.07831617  0.10879288
##  [4,] -0.01432345 -0.1731285  0.13542785 -0.09439452  0.06442843
##  [5,] -0.04438047 -0.1809317  0.08200538 -0.11011924  0.02341834
##  [6,] -0.07443748 -0.1867759  0.03009840 -0.12987347 -0.01718253
##  [7,] -0.10449449 -0.2018545 -0.01383219 -0.15456277 -0.05495744
##  [8,] -0.13455151 -0.2296262 -0.05079586 -0.17966164 -0.09066718
##  [9,] -0.16460852 -0.2613791 -0.08024071 -0.21190440 -0.11815379
## [10,] -0.19466554 -0.3008540 -0.09252488 -0.24832965 -0.14217330
## [11,] -0.22472255 -0.3516228 -0.10185277 -0.28734252 -0.16254720
```

Let us plot these results.

```r
plot(ideo.table[1:11, 1], ylim = c(-1, 1), axes = FALSE, type = "l", col = "black",
    lwd = 2, lty = 1, xlab = "# of Correct Answers", ylab = "Marginal Effect of Ideology",
    main = "Conditional Effects of Ideology")
axis(1, at = c(1:11), labels = c(0:10))
axis(2, at = c(-1, -0.5, 0, 0.5, 1))
lines(c(1:11), rep(0, 11))
matlines(1:11, ideo.table[1:11, c(2, 3)], col = "gray70", lty = 2, lwd = 2)
```




```r
plot(c(1:11), c(ideo.table[1:11, 1]), axes = FALSE, ylim = c(-1, 1), xlab = "# of Correct Answers",
    ylab = "Marginal Effect of Ideology", main = "Conditional Effects of Ideology")
axis(1, at = c(1:11), labels = c(0:10))
axis(2, at = c(-1, -0.5, 0, 0.5, 1))
lines(c(1:11), rep(0, 11))
segments(c(1:11), c(ideo.table[, 2]), c(1:11), c(ideo.table[, 3]), lwd = 1,
    lty = 2)
segments(c(1:11), c(ideo.table[, 4]), c(1:11), c(ideo.table[, 5]), lwd = 2,
    lty = 1)
```




More.

```
X.tilde <- cbind(rep(1, 5), rep(1, 5), seq(1, 5, length = 5), rep(5, 5), rep(30,
    5), rep(0, 5), rep(0, 5), rep(0, 5), rep(0, 5), rep(8, 5))

n.tilde <- 5

n.sims <- 1000
sim.model1 <- sim(model1, n.sims)

y.tilde <- array(NA, c(n.sims, n.tilde))

for (i in 1:n.sims) {
    y.tilde[i, ] <- rnorm(n.tilde, X.tilde %*% coef(sim.model1)[i, ], sigma.hat(sim.model1)[i])
}


p.ideo <- matrix(NA, nrow = 5, ncol = 1)
for (i in 1:5) {
    p.ideo[i, 1] <- mean(y.tilde[, i])
}

p.ideo.ci <- matrix(NA, nrow = 5, ncol = 4)
for (i in 1:5) {
    p.ideo.ci[i, ] <- quantile(y.tilde[, i], prob = c(0.025, 0.16, 0.84, 0.975))
}
```

```r
ideo.table <- cbind(p.ideo, p.ideo.ci)
ideo.table
```

```
##          [,1]        [,2]      [,3]     [,4]     [,5]
## [1,] 2.705634  0.02324384 1.3352804 4.118152 5.406183
## [2,] 2.560727  0.01084824 1.1704884 3.951051 5.118429
## [3,] 2.434099 -0.26989106 1.0629342 3.854745 5.303145
## [4,] 2.282214 -0.24745311 0.8957789 3.629944 5.073984
## [5,] 2.082858 -0.54154529 0.7448645 3.441931 4.748018
```

# 4. Regression discontinuity designs

### Remember

```r
fun = c("R", "is", "fun")
paste(fun, collapse = " ")
```

```
## [1] "R is fun"
```