Lab Session: Regression Discontinuity Design

Cristian Huse

Introduction

This lab session is about **Regression Discontinuity Design (RD)**. The structure of the session is as follows:

- Preparation;
  - Load libraries, setwd, load data (.dta format)
- Example 12. Regression Discontinuity Design Estimates
- Nonlinear Functions Around the Threshold
- Narrowing the Window Around the Threshold

While we also briefly discuss the fuzzy RD design, the focus of the empirical exercise is on the sharp RD design.

Please note that our discussion and estimation are based on first principles. See, e.g. https://rdpackages.github.io/ for packages and more advanced material.

Preparation

```
## Initialize ####

rm(list=ls())

#Load libraries
library(fixest) #for feols etc
library(haven) #for read_dta
library(modelsummary) #for neat tables
library(tidyverse)

## Initialize ####
## Set working directory
#Specify the access path to the computer folder you will use for the analysis
#setwd("INSERT PATH OF FOLDER WHERE YOU SAVED THE DATASET WITH / or \\")
setwd("C:/Users/huse-admin/Dropbox/CRISTIAN/Teaching/Cursos_Meus/Teaching_2021/EPE/Lab5_RDD")

## open data
#Open the cleaned data set
#set path for data
```

```r
evaluation <- file.path(getwd(), "Data", "evaluation.dta")
#import .dta file
library(haven) #for read_dta
evaluation.df <- read_dta(evaluation)
```

Regression Discontinuity Design

As discussed in the lectures, many programmes base eligibility according to a continuous variable (**running variable**); only individuals above/below a given **threshold** (cut-off) are eligible to participate in the programme. This is a setting where regression discontinuity (RD) can be used.

Denoting the running variable by X_i, the threshold by X_0 and assume eligibility is determined by being below the threshold, so $P_i = 1(X_i \leq X_0)$. A so called sharp RD design occurs when all units below or above have the same treatment status, without exceptions. The associated regression to be estimated takes the form

$$Y_i = \beta + \delta P_i + f(X_i) + \varepsilon_i$$

where Y_i is the outcome of interest for individual i, P_i is the eligibility indicator for individual i, $f(X_i)$ is a continuous function around the threshold, so that the value of this function tends to be asymptotically equal around the threshold, and ε_i denotes a random error term.

The use of a continuous function of the running variable occurs to account for nonlinearities in the relationship between the running variable and the outcome. To appreciate the importance of the continuity of the function, compare the outcomes of individuals at the cut-off (index 0) against individuals just below the cut-off (index 1):

$$Y_{i0} = \beta + \delta \times 0 + f(X_0) + \varepsilon_{i0}$$
$$Y_{i1} = \beta + \delta \times 1 + f(X_0 - \tau_i) + \varepsilon_{i1}$$

The difference between the above is

$$Y_{i1} - Y_{i0} = \delta + (f(X_0 - \tau_i) - f(X_0)) + (\varepsilon_{i1} - \varepsilon_{i0})$$

The continuity of $f()$ around the cut-off X_0 implies that $(f(X_0 - \tau_i) - f(X_0))$ vanishes as one gets closer to the cut-off. By taking expectations, $(\varepsilon_{i1} - \varepsilon_{i0})$ also vanishes, thus the **local average treatment effect (LATE)** at the threshold is estimated by δ.

While we will not treat the fuzzy RD design in this session, its intuition is that since the eligibility threshold does not fully determine participation in the programme, one needs to **predict** participation, which is done through the IV framework. On the first-stage, the idea is to predict participation by regressing P_i (**participation**) on an indicator $1(X_i \leq 0)$ (**eligibility**), i.e., $P_i = \gamma_0 + \gamma_1 1(X_i \leq 0) + \eta_i$. That is, although the indicator does not fully determine whether an individual will participate in the programme or not, it strongly influences programme participation, therefore being used as an instrumental variable to predict program participation.

On the second stage, the idea is to estimate a version of the standard RD regression with $\hat{P}_i$ replacing P_i, which again yields the **LATE**.

Example 12. Regression Discontinuity Design Estimates

Back to the HISP case study, assume that eligibility for the programme depends on a proxy poverty index and that data for the poverty index are available only in localities where the program will be offered. Households with a score below a certain cut-off (in this case, a value of 58) are chosen to participate in the program. Households with a score above that cut-off do not participate. The key assumption leading to the use of the sharp RD design is that the eligibility rule is strictly enforced, without any exceptions of either side of the cut-off. As before, the outcome of interest is health expenditures, which is measured after the end of the program. What we will do next is to compare health expenditures at follow-up between households just above and just below the poverty index threshold, in the treatment localities.

To implement the RD design, first normalize the poverty index threshold to 0 and create dummy variables for households with a poverty-targeting index to the left or right of the threshold. This allows the relationship between the outcome variables and the running variable (poverty index) to have different slopes on either side of the threshold.

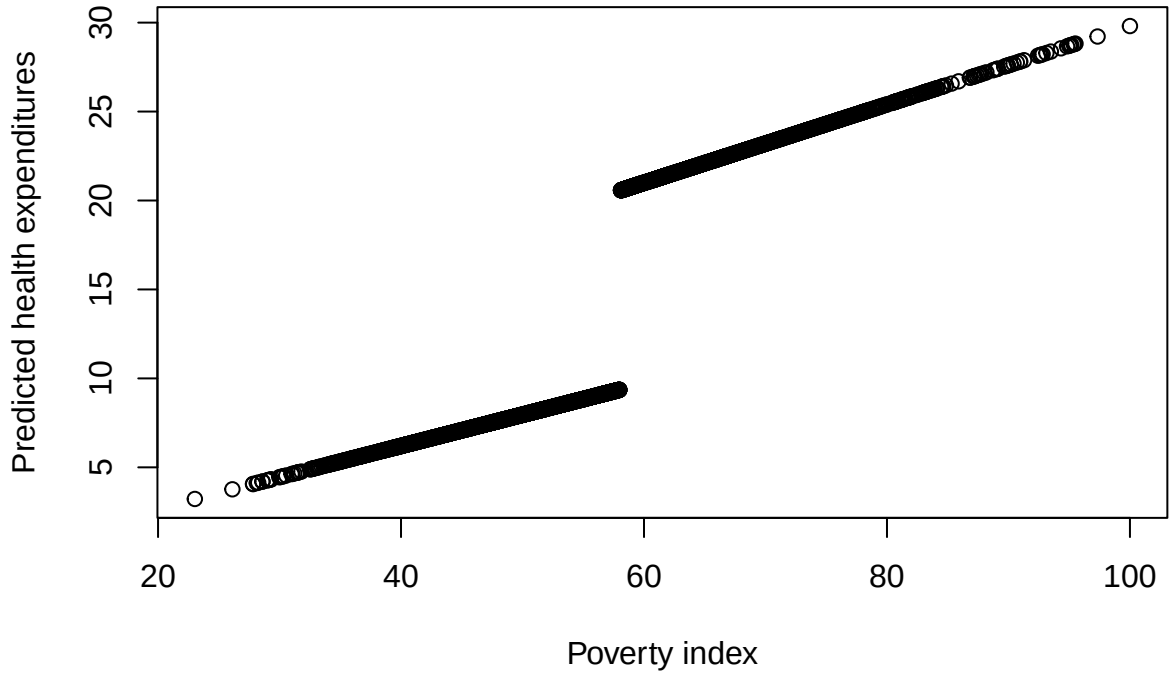
Next, run a regression of health expenditures on the indicator of exposure to the programme, as well as the two dummies for whether households have a poverty index to the left or to the right of the threshold. The resulting RD estimate for δ is -11.19.

```
# Select the relevant data and normalize the poverty index
rd.df <- subset(evaluation.df, treatment_locality == 1 & round == 1)
rd.df$poverty_index_left<-ifelse(rd.df$poverty_index <= 58,rd.df$poverty_index - 58, 0)
rd.df$poverty_index_right<-ifelse(rd.df$poverty_index > 58,rd.df$poverty_index - 58, 0)

ex12_lm1 <- lm(health_expenditures ~ poverty_index_left + poverty_index_right +
                 eligible,
               data = rd.df)

ex12_feols1 <- feols(health_expenditures ~ poverty_index_left +
                       poverty_index_right + eligible,
                     cluster = ~ locality_identifier,
                     data = rd.df)

models12 <- list("ex12_lm1" = ex12_lm1, "ex12_feols1" = ex12_feols1)

modelsummary(models12,
             vcov = ~ locality_identifier,
             stars = c("*" = .1, "**" = .05, "***" = .01),
             fmt = 3,
             gof_omit = "AIC|BIC|Log.Lik.|R2 Adj.|R2 Within|R2 Pseudo|F")
```

```
#Creating a simple graph
rd.df$ex12_lm1_pred <- ex12_lm1$fitted.values
plot(rd.df$poverty_index, rd.df$ex12_lm1_pred,
     xlab = "Poverty index", ylab = "Predicted health expenditures")
```

	ex12_lm1	ex12_feols1
(Intercept)	20.554***	20.554***
	(0.495)	(0.495)
poverty_index_left	0.176***	0.176***
	(0.037)	(0.037)
poverty_index_right	0.220***	0.220***
	(0.052)	(0.052)
eligible	−11.192***	−11.192***
	(0.581)	(0.581)
Num.Obs.	4960	4960
R2	0.338	0.338
Std.Errors	C: locality_identifier	C: locality_identifier

* p < 0.1, ** p < 0.05, *** p < 0.01



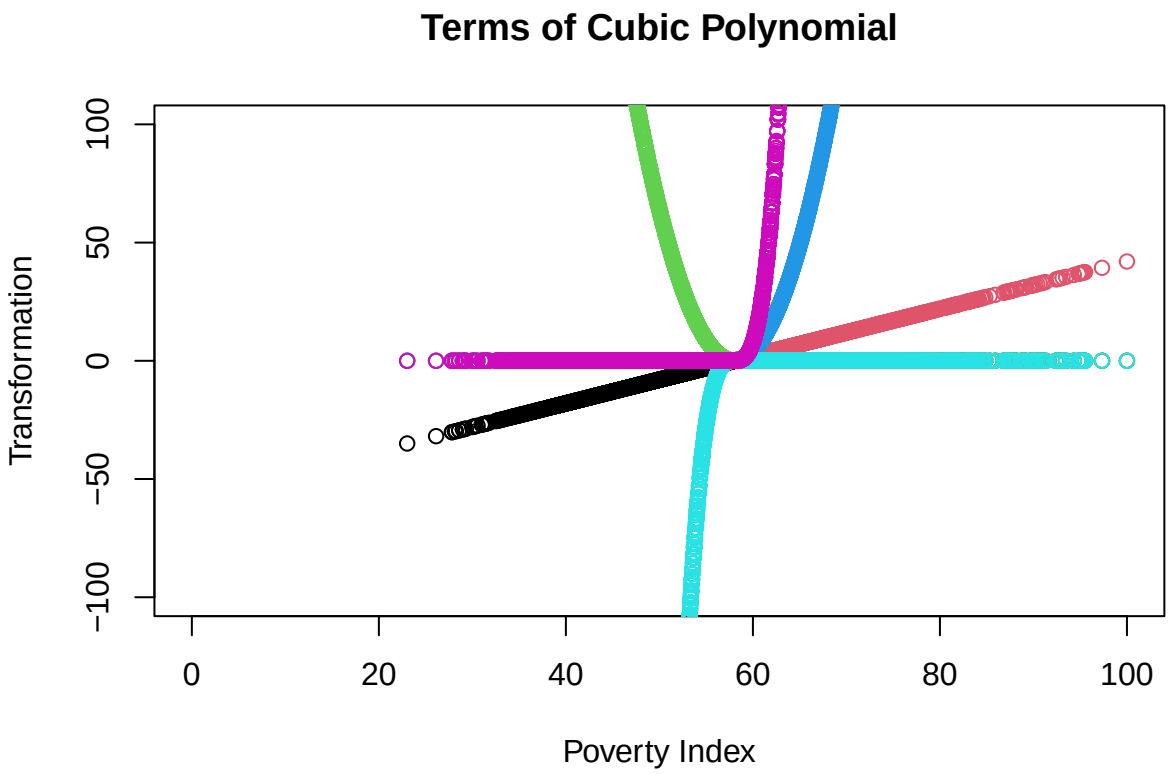
Nonlinear Functions Around the Threshold

One natural question to ask yourself is what would happen if $f()$ is actually nonlinear and we didn't account for it. In what follows, we define $f()$ as a cubic polynomial and provide a rough plot (individual ones are commented out) of the polynomial terms (note that we allow left and right terms to have different coefficients in the regression).

Bottom-line is, that since the coefficients associated to the higher order terms of the polynomials aren't statistically significant, the effect of the program doesn't change substantially; at −10.59, its effect is quite

similar to the original estimates of −11.19.

```
rd.df$poverty_index_left2<-ifelse(rd.df$poverty_index <= 58,(rd.df$poverty_index - 58)^2, 0)
rd.df$poverty_index_right2<-ifelse(rd.df$poverty_index > 58,(rd.df$poverty_index - 58)^2, 0)

rd.df$poverty_index_left3<-ifelse(rd.df$poverty_index <= 58,(rd.df$poverty_index - 58)^3, 0)
rd.df$poverty_index_right3<-ifelse(rd.df$poverty_index > 58,(rd.df$poverty_index - 58)^3, 0)

# Terms of polynomial
#summary(rd.df$poverty_index_left,rd.df$poverty_index_right,rd.df$poverty_index_left2, rd.df$poverty_in

#plot(rd.df$poverty_index, rd.df$poverty_index_left,      xlim=c(0,100), ylim=c(-100,100), xlab = "Pove
#points(rd.df$poverty_index, rd.df$poverty_index_right, col = 2)

#plot(rd.df$poverty_index, rd.df$poverty_index_left2, col = 3, xlim=c(0,100), ylim=c(-100,100), xlab =
#points(rd.df$poverty_index, rd.df$poverty_index_right2, col = 4)

#plot(rd.df$poverty_index, rd.df$poverty_index_left3, col = 5, xlim=c(0,100), ylim=c(-100,100), xlab =
#points(rd.df$poverty_index, rd.df$poverty_index_right3, col = 6)

#Combined
plot(rd.df$poverty_index, rd.df$poverty_index_left,
     xlim=c(0,100), ylim=c(-100,100),
     xlab = "Poverty Index", ylab = "Transformation",
     main = "Terms of Cubic Polynomial")
points(rd.df$poverty_index, rd.df$poverty_index_right, col = 2)
points(rd.df$poverty_index, rd.df$poverty_index_left2, col = 3)
points(rd.df$poverty_index, rd.df$poverty_index_right2, col = 4)
points(rd.df$poverty_index, rd.df$poverty_index_left3, col = 5)
points(rd.df$poverty_index, rd.df$poverty_index_right3, col = 6)
```




```
ex12_lm2 <- lm(health_expenditures ~ poverty_index_left + poverty_index_right +
                 poverty_index_left2 + poverty_index_right2 +
                 poverty_index_left3 + poverty_index_right3 +
                 eligible,
               data = rd.df)

ex12_feols2 <- feols(health_expenditures ~ poverty_index_left + poverty_index_right +
                       poverty_index_left3 + poverty_index_right3 +
                       eligible,
                     cluster = ~ locality_identifier,
                     data = rd.df)

models12 <- list("ex12_lm1" = ex12_lm1, "ex12_feols1" = ex12_feols1,
                 "ex12_lm2" = ex12_lm2, "ex12_feols2" = ex12_feols2)

modelsummary(models12,
             vcov = ~ locality_identifier,
             stars = c("*" = .1, "**" = .05, "***" = .01),
             fmt = 3,
             gof_omit = "AIC|BIC|Log.Lik.|R2 Adj.|R2 Within|R2 Pseudo|F")
```

	ex12_lm1	ex12_feols1	ex12_lm2	ex12_feols2
(Intercept)	20.554***	20.554***	19.779***	19.779***
	(0.495)	(0.495)	(0.736)	(0.735)
poverty_index_left	0.176***	0.176***	0.108	0.108
	(0.037)	(0.037)	(0.162)	(0.162)
poverty_index_right	0.220***	0.220***	0.561**	0.561**
	(0.052)	(0.052)	(0.238)	(0.238)
eligible	−11.192***	−11.192***	−10.594***	−10.594***
	(0.581)	(0.581)	(0.861)	(0.860)
poverty_index_left2			−0.006	−0.006
			(0.014)	(0.014)
poverty_index_right2			−0.029	−0.029
			(0.021)	(0.021)
poverty_index_left3			0.000	0.000
			(0.000)	(0.000)
poverty_index_right3			0.001	0.001
			(0.000)	(0.000)
Num.Obs.	4960	4960	4960	4960
R2	0.338	0.338	0.339	0.339
Std.Errors	C: locality_identifier	C: locality_identifier	C: locality_identifier	C: locality_identifier

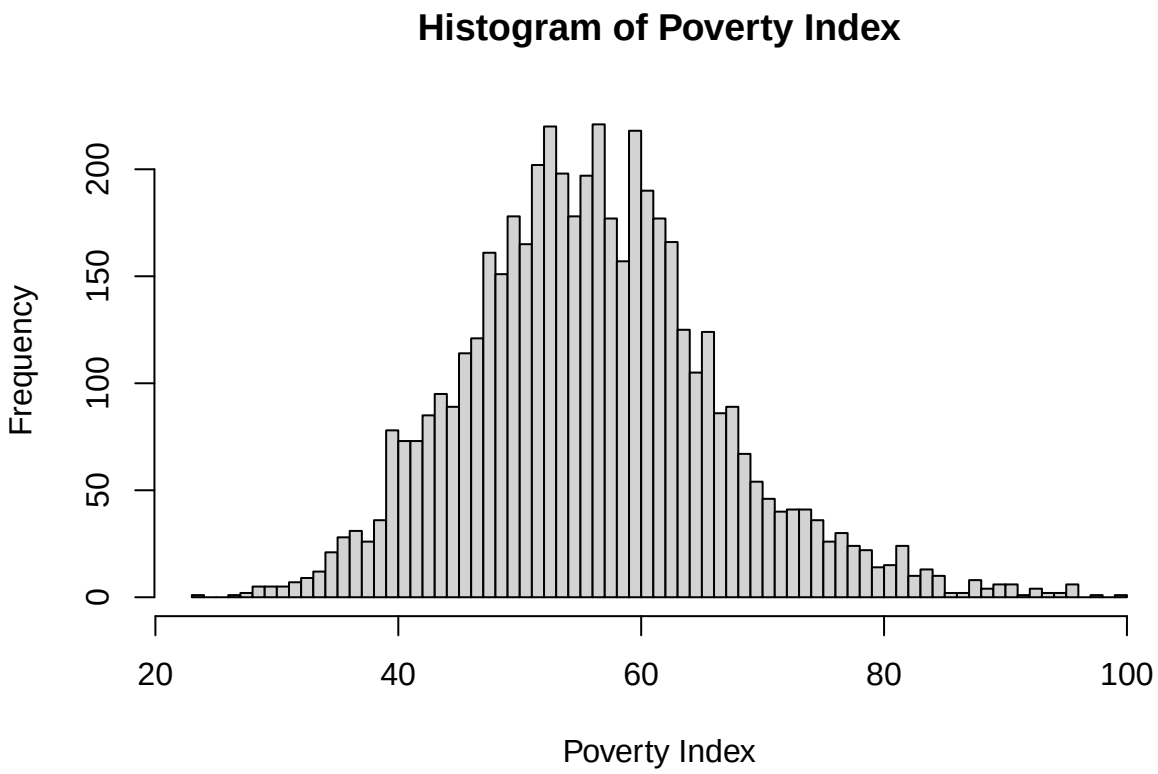
* p < 0.1, ** p < 0.05, *** p < 0.01

Narrowing the Window Around the Threshold

We now illustrate another dimension of RD estimates. Recall that identification in the RD design relies on a conditional expectation as one approaches the threshold, i.e., it relies on a mass of observations just above and just below the threshold. That is, **the window width is crucial for identification**. However, by dramatically reducing the window width one is decreasing the sample size effectively used in the analysis, leading to less price estimates. In other words, the choice of window width entails – as in many other cases the famous **bias-variance trade-off**.

In what follows, we plot a (very undersmoothed) histogram of the poverty index to get a sense of how the variable looks like. We then estimate a number of models whereby we progressively narrow the window around the threshold from full sample down to +-40, +-30, +-20, +-10, +-5 up to +-2. Narrowing the window obviously drops observations from the estimation sample (see table), but you will realize that the estimates are largely robust up to +-5 around the threshold. Also expected is the increase in standard errors due to the smaller sample. All in all, the results are largely robust to changes in window width and you can see the **bias-variance trade-off** in practice. The table reports a subset of specifications.

```
# This is undersmoothed (= many classes) on purpose
hist(rd.df$poverty_index, breaks = 75, main = "Histogram of Poverty Index",
     xlab = "Poverty Index")
```




```
summary(rd.df$poverty_index)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.03   49.07   55.51   56.01   62.06  100.00
```

```
#range: 23.03-100

ex_lm_full <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                     data = rd.df)
# Window +-40
rd.df$window_pm40 <- (rd.df$poverty_index<=98&rd.df$poverty_index>=18)
#sum(rd.df$window_pm40)
ex_lm_40 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm40,
                     data = rd.df)
# Window +-30
rd.df$window_pm30 <- (rd.df$poverty_index<=88&rd.df$poverty_index>=28)
#sum(rd.df$window_pm30)
ex_lm_30 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm30,
                     data = rd.df)
# Window +-20
```

```
rd.df$window_pm20 <- (rd.df$poverty_index<=78&rd.df$poverty_index>=38)
#sum(rd.df$window_pm20)
ex_lm_20 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm20,
                     data = rd.df)
# Window +-10
rd.df$window_pm10 <- (rd.df$poverty_index<=68&rd.df$poverty_index>=48)
#sum(rd.df$window_pm10)
ex_lm_10 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm10,
                     data = rd.df)
# Window +-5
rd.df$window_pm5 <- (rd.df$poverty_index<=63&rd.df$poverty_index>=53)
#sum(rd.df$window_pm5)
ex_lm_5 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm5,
                     data = rd.df)
# Window +-2
rd.df$window_pm2 <- (rd.df$poverty_index<=60&rd.df$poverty_index>=56)
#sum(rd.df$window_pm2)
ex_lm_2 <- lm(health_expenditures ~ eligible +
                 poverty_index_left + poverty_index_right,
                    subset = rd.df$window_pm2,
                     data = rd.df)
```

```
#Report some of them
modelswindow <- list("Full Sample" = ex_lm_full, "Window: +- 30" = ex_lm_30,
                     "Window: +- 10" = ex_lm_10, "Window: +- 2" = ex_lm_2)

modelsummary(modelswindow,
             vcov = ~ locality_identifier,
             stars = c("*" = .1, "**" = .05, "***" = .01),
             fmt = 3,
             gof_omit = "AIC|BIC|Log.Lik.|R2 Adj.|R2 Within|R2 Pseudo|F")
```

References

Gertler, Paul J.; Martinez, Sebastian; Premand, Patrick; Rawlings, Laura B.; Vermeersch, Christel M. J. (2016). Impact Evaluation in Practice, Second Edition, Technical Companion (Version 1.0). Washington, DC: Inter-American Development Bank and World Bank.

	Full Sample	Window: +- 30	Window: +- 10	Window: +- 2
(Intercept)	20.554***	20.693***	20.018***	17.683***
	(0.495)	(0.484)	(0.637)	(1.017)
eligible	−11.192***	−11.341***	−10.799***	−7.909***
	(0.581)	(0.579)	(0.750)	(1.267)
poverty_index_left	0.176***	0.174***	0.140*	0.379
	(0.037)	(0.037)	(0.078)	(0.752)
poverty_index_right	0.220***	0.197***	0.370***	2.373***
	(0.052)	(0.053)	(0.123)	(0.890)
Num.Obs.	4960	4923	3324	773
R2	0.338	0.334	0.294	0.230
Std.Errors	C: locality_identifier	C: locality_identifier	C: locality_identifier	C: locality_identifier

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$