Regression Discontinuity Design

```
knitr::opts_chunk$set(
    echo = TRUE,
    message = FALSE,
    warning = FALSE,
    comment = NA
)

rm(list = ls())

library(readr)
educ <- read_csv("islamic_women.csv")
```

Lecture Slides & Data

```
#install.packages("downloadthis")
library(downloadthis)

download_link(
  link = "https://bayreuth-politics.github.io/CI22/data/islamic_women.csv",
  output_name = "educ",
  output_extension = ".rdata",
  button_label = "Lab 8 Data",
  button_type = "success",
  has_icon = TRUE,
  self_contained = TRUE
)

#download_link(
#  link = "https://github.com/dpir-ci/CI22/raw/gh-pages/docs/lectures/lecture7.pdf",
#  output_name = "week7",
#  output_extension = ".pdf",
#  button_label = "Lecture Slides",
#  button_type = "default",
#  has_icon = FALSE,
#  self_contained = FALSE
#)
```

Recap

Sharp Regression Discontinuity

In RDDs we exploit that treatment assignment is determined by a known **assignment rule** that determines whether units are assigned to the treatment. In RDD, all units in the study receive a score, usually called *running variable*, *forcing variable* or *index*, and treatment is assigned to those units whose score is above a known cut-off (Cattaneo et al 2019).

In RDD we have three components:

1. The score
2. The cutoff
3. The treatment.

In the Sharp Regression Discontinuity Design, the score determines - deterministically - whether the unit is being assigned to treatment or to the control condition. However, we again face the fundamental problem of causal inference because we can only observe the untreated outcome for those units below the cutoff (control) and the treated outcome for those above the cut-off (treated). However, imposing the assumption of comparability between units with very similar values of the score but on opposite sides of cut-off enables us to calculate the treatment effect of the intervention.

Given the continuity assumption, we would expect that observations in a small neighbourhood around the cut-off will have very similar potential outcomes. Thus, this would justify using observations just below the cut-off as a reasonable proxy of what would the average outcome for those units just above the cut-off would have had if they had received the control condition instead of the treatment (i.e. counterfactual).

The main goal in RD is to adequately perform an extrapolation of the average outcome of treated and untreated units at the cutoff.

Estimation

There are different ways that we could estimate the causal effect from RDD:

- Linear
- Linear with different slopes

- High-order polynomial (or non-linear)
- Non-parametric

Choice of Kernel function and Polynomial Order

Whenever we conduct RDD we choose the following:

1. Polynomial order p function
2. Kernel function $K(\dot{})$, which determines how observations within a bandwidth would be weighted, and there are different options: "Uniform", "Triangular", and "Epanechnikov".
3. Bandwidth size: There is a trade-off between bias and efficiency. The closer you get to the cut-off, the less bias in the estimator, but more variance as there are fewer observations.
4. We conduct weighted least squares and obtain the intercept of the chosen polynomial order above and below the cut-off.
5. Calculate the sharp point estimate by subtracting the intercepts of these polynomials: $\tau_{\widehat{SRD}} = \alpha_{above} - \alpha_{below}$

Falsification tests:

Whenever we conduct RD we choose the following:

We learned that we can conduct multiple sensitivity and falsification tests:

1. Sensitivity: To check that the results of our estimation are robust to different specifications
2. Continuity: To check whether covariates do not jump at the threshold.
3. Sorting: To check that units do not sort around the threshold.
4. Placebo cut-offs: To check that outcomes do not change abruptly at an arbitrary threshold (to test the continuity assumption)

Before starting this seminar

1. Create a folder called "lab8"

2. Download the data (you can use the button or the one at the top, or read csv files directly from github):

3. Open an R script (or Markdown file) and save it in our "lab8" folder.

4. Set your working directory using the setwd() function or by clicking on “More“. For example *setwd(“~/Desktop/Causal Inference/2022/Lab8”)*

5. Let’s install an load packages that we will be using in this lab:

```r
library(stargazer) # generate formated regression tables
library(texreg) # generate formatted regression tables
library(tidyverse) # to conduct some tidy operations
library(ggplot2)
#install.packages(c(rdrobust, rddensity))
library(tidyverse)
#install.packages("rdd")
library(rdd) # to conduct McCrary Sorting Test
library(rdrobust) # to conduct non-parametric rd estimates
library(rddensity) # to conduct a density test and density plots
```

Seminar Overview

In this **seminar**, we will cover the following topics:
1. Conduct regression discontinuity using global polynomial estimation using `lm()` function
2. Calculate LATE using non-parametric estimations using `rdrobust()`.
3. Conduct various robustness/falsification tests such as balance test, placebo outcome, density test, and falsification test.

Islamic Rule and the Empowerment of the Poor and Pious - Meyerson (2014)

In this paper, Meyerson is interested in the effects of Islamic parties' control of local governments on women's rights. He focuses on the educational attainment of young women. Meyerson conducts a Sharp RD design, based on close elections in Turkey. The challenge here is to compare municipalities where the support for Islamic parties is high and win the election, versus those that elected a secular mayor.

You would expect that municipalities controlled by Islamic parties would systematically differ from those that are controlled by a secular mayor. Particularly, if religious conservatism affects the educational outcomes of women. However, we can use RDD to isolate the treatment effect of interest from all systematic differences between treated and untreated units.

We can compare municipalities the Islamic party barely won the election versus municipalities where the Islamic party barely lost. This reveals the causal (local) effect of Islamic party control on women's educational attainment a few years later. One crucial condition to meet in this setup is that parties cannot systematically manipulate the vote share they obtain.

The data used in this study is from the 2014 mayoral election in Turkey. The unit of analysis is the municipality, and the running variable is the margin of victory. The outcome of interest is the educational attainment of women who attended high school during 1994-2000, calculated as a percentage of the cohort of women aged 15 to 20 in 2000 who had completed high school by 2000.

We will be using the following variables:

Variable	Description
`margin`	This variable represents the margin of victory of Islamic parties in the 1994 election. A positive margin means that an Islamic party won.
`school_men`	secondary school completion rate for men aged between 15 and 20
`school_women`	the secondary school completion rate for women aged 15-20
`log_pop`	log of the municipality population in 1994
`sex_ratio`	gender ratio of the municipality in 1994
`log_area`	log of the municipality area in 1994

Now let's load the data. There are two ways to do this:

You can load the dataset from your laptop using the `read.csv()` function. Here the dataset is called `educ` - but feel free to give it a different name if you prefer.

```
# Set your working directory
#setwd("~/Desktop/Causal Inference/2022/Lab8")
#
library(readr)
#educ <- read.csv("~/islamic_women.csv")

head(educ)
```

Let's start by visualising the data. We will use `plot()` function to do this. This is the simplest scatter plot that you can come up with.

Exercise 1: Generate a plot using the `plot(X,Y)` function. Replace X with `educ$margin` and Y with `educ$school_women`.

Reveal Answer

```
plot(educ$margin, educ$school_women)
```

This is a very simple plot that shows the raw relationship between the margin of victory and the outcome variable. However, it conveys some important information. For example, the margin of victory is clustered around -0.5 and roughly 0.3. Also, the outcome variable, school attainment, usually goes from 0 to 40%. Now let's generate a slightly fancier plot.

Exercise 2: Generate a scatter plot using `ggplot()` function. Use the functions below to add some additional features into this plot.

```
ggplot(aes(x = running variable, y = outcome, colour =outcome), data = data) +
  # Make points small and semi-transparent since there are lots of them
  geom_point(size = 0.5, alpha = 0.5, position = position_jitter(width = 0, height = 0.25,
  # Add vertical line
  geom_vline(xintercept = 0) +
  # Add labels
  labs(x = "Label X", y = "Label Y") +
  # Turn off the color legend, since it's redundant
  guides(color = FALSE)
```

Reveal Answer

```
# Let's check if this is a sharp RD.
ggplot(educ, aes(x = margin, y = school_women, colour =school_women)) +
  # Make points small and semi-transparent since there are lots of them
  geom_point(size = 0.5, alpha = 0.5,
             position = position_jitter(width = 0, height = 0.25, seed = 1234)) +
  # Add vertical line
  geom_vline(xintercept = 0) +
  # Add labels
  labs(x = "Vote share", y = "Womens' Educational Attainment (Proportion)") +
  # Turn off the color legend, since it's redundant
  guides(color = FALSE)
```

We can now see more clearly that most municipalities elected a secular mayor. Recall that we are using the margin of victory for the Islamic parties - a positive margin of victory is positive means that the Islamic parties won the election in that municipality.

Exercise 3: Let's generate a dummy variable that is equal to "Treated" if the margin of victory *is equal or greater than zero* and "Untreated" otherwise. You can use the `mutate()` function and the `ifelse()` functions to do this. Remember to use the pipeline operator to store the variable into your existing data frame. See below the syntax for more information.

Function/argument	Description
`data <- data %>%`	Pipeline operator to assign the new operation into a new data or existing data frame
`mutate(new variable = ifelse(variable >= "condition", "Treated", "Untreated")`	If the condition is met, the new variable takes value equal to “Treated” and “Untreated” otherwise

```
data <- data %>%
  mutate(newvariable = ifelse(variable >= 0 , "Treated", "Untreated"))
```

Reveal Answer

```
educ <- educ %>%
  mutate(treat = ifelse(margin >= 0 , "Treated", "Untreated"))
```

Now that we have created our treatment condition variable, let’s generate an additional plot that conveys information on the distribution of the running variable for both treated and untreated municipalities.

Exercise 4: Generate a plot looking at the distribution margin of victory variable. Using the `ggplot()` function, set the `x` argument equal to the margin of victory variable. Set `fill` equal the new treatment variable `treat`. There is no need to add the `y` argument given that we expect to generate a histogram that will give us the number of observations for each value in the margin of victory variable. Add the `geom_histogram()` function. Set the argument `binwidth` in this function equal to 0.01 and set `colour` argument equal to “dark”. Let’s add the `geom_vline()` function and add a vertical line by setting the `xintercept` argument equal to zero. Add the `labs()` function and set `x` label equal to “Margin of victory” and the `y` label equal to “count”, and the `fill` argument equal to “Treatment Status”.

Reveal Answer

```r
ggplot(educ, aes(x = margin, fill = treat)) +
  geom_histogram(binwidth = 0.01, color = "white") +
  geom_vline(xintercept = 0) +
  labs(x = "Margin of Victory", y = "Count", fill = "Treatment Status")
```

In this plot, we can see the distribution of the running variable (margin of victory). Again, we can see that in the majority of the municipalities a secular mayor was elected.

Global Parametric Estimation

As we discussed in the lecture, there are different ways to estimate the causal effect using RD. These different approaches differ in the range of observations they include as well as how they estimate the average outcome for those units just above the cut-off and below the cut-off. One way to estimate the effect of the intervention in RDD is using OLS, but only using the running variable as the main predictor. In this case, we use the running variable measured as the distance from the cut-off. Stated formally: $\tilde{X}_l = X - c$.

In this case, the regression in the left-hand side would be equal to:

$$Y = \alpha_l + \tilde{X} + \epsilon$$

Whereas the regression above the cut-off is equal to:

$$Y = \alpha_r + \tilde{X} + \epsilon$$

It's important to point out that for all estimations the treatment effect is equal to the differences of the intercepts of the regressions above and below the cut-off.

$$\tau = \alpha_r - \alpha_l$$

Let's do this manually. To do so, subset the data for those observations above and below the threshold. Then, regress the outcome on the **running variable**. Finally, subtract the intercepts from each regression. Let's do that in the following exercise.

Exercise 5: Run a regression using only `margin` variable as the predictor. Set your outcome variable equal to `school_women` variable. Set the `data` argument equal

to `educ` (unless you called your data frame differently). Add the `subset` function inside of the `lm()` function and set it equal to `margin >= 0` for the regression above the cut-off and `margin < 0` for the regression below the cut-off. Use the `summary()` function to report your results, but also store the output of this function into an object. You can retrieve the intercept from the object that you stored the output from the `summary()` function this way:

```
object <- summary(lm(outcome ~ variable, data = data, subset = variable >= condition))

# intercept
object$coefficient[1]
```

Reveal Answer

```
# above
above <- summary(lm(school_women~margin, data = educ, subset = margin >= 0))

above$coefficients[1]

# above
below <- summary(lm(school_women~margin, data = educ, subset = margin < 0))

below$coefficients[1]

# 0.156453 - 0.162002
tau_rd = above$coefficients[1] - below$coefficients[1]
tau_rd
```

Based on this approach the intercepts of the two regressions yield the estimated value of the average outcome at the cut-off point for the treated and untreated units. This difference of intercepts is the estimated effect of an Islamic party being in power (at the municipal level) - it suggests there is a 0.5 percentage decrease in women's educational attainment.

A more direct way of estimating the treatment effect is to run a pooled regression on both sides of the cut-off point, using the following specification: $Y = \alpha + \tau D + \beta \tilde{X} + \epsilon$

Where τ is the coefficient of interest. Here again LATE is the difference between the two intercepts: $\tau = \alpha_r - \alpha_l$. When D switches off and we are also controlling the different values of the forcing variable, $\tilde{X}$, we get the slope of the regression below the threshold. Conversely, for units above the cut-off, D switches on, and we control for different values of the forcing variable, we get the slope of the regression above the cut-off. The estimated effect of the treatment at $\tilde{X}$ then provides the treatment effect (τ).

Note that you are constraining the slope of the regression lines to the same on both sides of the cut-off. ($\beta_l = \beta_r$) This might not be consistent if the data structure varies and the single slope fails to appropriately approximating each side to the cutoff.

Exercise 6: Calculate the effect of the intervention using a regression model including the `margin` and `treat` variables. Use the`lm()` function to conduct this analysis. Store this regression into an object and call it `global2`. Use the `summary()` to inspect your results. Interpret the coefficient of interest.

Reveal Answer

```
# a more direct way is to run a pooled regression on both sides of the cut-off (constraini

global2 <- lm(school_women~treat+margin, data = educ)
summary(global2)
```

Using this estimation approach, we obtain that, on average, the effect of a municipality in control of an Islamic party leads to a 1.7%-point increase in women's educational attainment.

We can also allow the regression function to differ on both sides of the cut-off by including interaction terms between D and $\tilde{X}$. This would be as follows:

$$Y = \alpha_l + \tau D + \beta_0 \tilde{X} + \beta_1 (D \times \tilde{X}) + \epsilon$$

Let's do that in the following exercise.

Exercise 7: Calculate the effect of the intervention using a regression model including the `margin` and `treat` variables and the interaction between these two variables. Use the`lm()` function to conduct this analysis. Store this regression into an object and call it `global3`. Use the `summary()` to inspect your results. Interpret the coefficient of interest.

Reveal Answer

```
# a more direct way is to run a pooled regression on both sides of the cut-off (constraini

global3 <- lm(school_women~treat+margin + treat*margin, data = educ)
summary(global3)
```

Here, we find that the coefficient of interest is positive (`0.005`) yet insiginifcant. This means that, based on this model, the effect of an Islamic party in control of the municipal government does not lead to a change in women's educational attainment.

Global parametric models have a severe shortcoming - they rely upon observations that are far away from the cut-off. Indeed, the evidence against using a global polynomial approach is quite substantial. According to Cattaneo et al (2019), this estimation technique does not provide accurate point estimators and inference procedures with good statistical properties.

Exercise 7 (no coding required): Think about how the global polynomial approach weights each observation when it calculates the coefficient of interest?

Reveal Answer

OLS will estimate τ based on all observations across the score. This means that the observations' very far from the cut-off weight is equal to that of very close ones'. In the worst-case scenario, if the observations are clustered far from the cut-off, the estimation of τ would be heavily influenced by those values rather than those close.

We can also use high-order polynomial to retrieve LATE. However, the evidence against using high-order polynomial seems to be quite robust see here for a discussion on high-order polynomials). In short, the issues with using high-order polynomials is that they leads to noisy estimates, they are sensitivity to the degree of the polynomial, and they have poor coverage of confidence intervals.

Exercise 8: Well - let's give it a try nonetheless. Conduct a third-order polynomial regression function. Include in this model the `treat` and the `margin` variables. Also, add the `margin` variable raised at the power of 2 and then at the power of 3. We also need to include the `I()` function or insulate function for the `margin` variable that is raised at the power of 2 and 3. The `I()` function insulates whatever is

inside this function. It creates a new predictor that is the product of the margin variable by itself two and three times. Store this output in an object and call that object `global4`. Then, use the `summary()` to check the results of this specification. Interpret the results.

Reveal Answer

```
global4 <- lm(school_women~ treat + margin+I(margin^2)+ I(margin^3) , data = educ)

summary(global4)
```

Using a high-order polynomial function, we find that womens' educational attainment decreases, on average, by roughly `2.1` per cent when an Islamic party is in power - yet the result is insignificant.

You might have reason to prefer parametric approaches - or deem them more appropriate in some cases. Then, you should not resort to a global model. It's more appropriate to only use observations that are close to the cut-off (above and below). Let's run the unconstrained model from **Exercise 7**, but this time we only use observations that are within 0.5 percentage points above and below the threshold.

Exercise 9: Run the same model used in `Exercise 7`, but subset your data taking only observations that are above and below 0.5 points from the threshold. Store the results from this regression into an object and call it `local`. Use the `summary()` to check your results. If you don't remember how to subset data in the `lm()` function. See the syntax below

```
lm(outcome ~ variable1 + variable2 + variable1 * variable2, data=data,
            subset=(running_variable>=-0.5 & running_variable<=0.5))
```

Reveal Answer

```
local <- lm(school_women ~ margin + treat + treat * margin, data=educ,
            subset=(margin>=-0.5 & margin<=0.5))
summary(local)
```

We can see that using a local polynomial function, we find that the effect of Islamic rule is inconclusive in this case.

It is important to stress that modern empirical work using RDDs empirical work employs local polynomial methods. In this case, we are estimating the average outcomes for treated and untreated units using observations that are near the cut-off. This approach tends to be more robust and less sensitive to boundary and overfitting problems. In local polynomial point estimation, we are still using linear regression, but within a specific bandwidth near the threshold. In the following section, we will look at how to use the approach using a non-parametric estimation strategy.

Non-Parametric Estimation

Let's now estimate the LATE using a non-parametric estimator. Conveniently, we can easily do so by using the `rdrobust` package. As the name indicates, the package allows us to estimate robust measurements of uncertainty such as standard errors and confidence intervals. It is based on theoretical and technical work by Calonico, Cattaneo and Titiunik. `rdrobust` estimates robust bias-corrected confidence intervals that address the problem of undersmoothing conventional confidence intervals face in RDDs. In other words, a small bias would be required for them to be valid, which might not be the case. Moreover, they also address the poor performance of (non-robust) bias-corrected confidence intervals.

As suggested by the authors and somewhat counter-intuitively, we therefore use the point estimate provided by the conventional estimator, but robust standard errors, confidence intervals and p-values to make statistical inferences.

The `rdrobust` command has the following minimal syntax. You can use a uniform bandwidth or specify two different ones. We will work with further arguments later. Note that you do not need to specify if the running variable is centred on the cut-off as you can manually specify the cut-off using the `c`-argument.

```
robust_model = rdrobust(data$running_var, data$dependent_var,
                        c=[cutoff], h=[bandwidth])
```

Exercise 10: Estimate the LATE using `rdrobust` with a bandwidth of 5% on either side of the cutoff. Interpret your result.

Reveal Answer

```
robust_5=rdrobust(educ$school_women, educ$margin, c=0, h=0.05)
summary(robust_5)
```

Our point estimate is `0.023`. That is, a victory of an Islamic party would be associated with an increase in the rate of women who complete school by `2.3%` - however, the p-value and confidence intervals indicate that the estimate is not statistically significant. Therefore, based on this model, we would conclude that winning an election does *not* have a an effect on women schooling.

A bandwidth of 5% seems about reasonable. But we should better check different ones, too. Let's see what happens if we halve the bandwidth.

Exercise 11: Estimate the same model as before with a bandwidth of 2.5%. Report and interpret your results.

Reveal Answer

```
robust_25=rdrobust(educ$school_women, educ$margin, c=0, h=0.025)
summary(robust_25)
```

Well, we can see that the number of observations used to estimate the effect has been reduced - which is reasonable and we should be aware of. The point estimate did not change much and, as before, we find that there is in fact no significant effect at the cutoff.

Bandwidths of 5% or 2.5% around the cutoff seem somewhat reasonable in this case - but so would several others. How can we know what bandwidth we should use to estimate our effect?

Recall the trade-off we are facing when choosing bandwidths that was discussed in the lecture: On the one hand we know that more narrow bandwidths are associated with less biased estimates - we rely on units that are indeed comparable: their distance in the running variable being as-if random the closer we get to the cut-off. On the other hand: the wider the bandwidths, the smaller the variance. As in several cases before, the structure of our data, such as the number of observations, plays an important role. Even if small bandwidths are desirable, it can be hard [impossible] to estimate a robust and significant effect if the number of observations around the cut-off is very small - even if there is a *true* effect.

Luckily, the `rdrobust` package provides a remedy for this. The packages allows us to specify that we want to use bandwidths that are optimal given the data input. The `rdrobust` command then picks the bandwidth that optimises the *mean square error* - in other words, *MSE-optimal* bandwidths. Note that this is the default bandwidth if you don't specify any. Let's try to find out what this would be in our case.

Exercise 12: Estimate the LATE using `rdrobust` and MSE-optimal bandwidths. To specify the model, replace the `h` argument with `bwselect="mserd"`.

Reveal Answer

```
robust_mserd=rdrobust(educ$school_women, educ$margin, c=0, bwselect="mserd")
summary(robust_mserd)
```

This now looks very different. We estimate a LATE of about `3%`, which is significant at the 90% lvel. Note that the optimal bandwidths has been estimated to be `17.2%-points` on either side of the cut-off, with a separate optimal bandwidth for bias correction. Note that the *MSE-optimal* bandwidth is optimal in statistical terms - we should always make sure to asses the bandwidths against our theory. Here, we'd compare parties' winning margins up to `17%-points`.

Note that, so far, we have used a single bandwidth for data below and above the cut-off. We can also specify different ones - both manually and in terms of optimal bandwidths. As the structure of our data might differ, different bandwidths might be optimal. We can specify two different *MSE-optimal* bandwidth selectors by specifying `bwselect="msetwo"`.

Exercise 13: Estimate the LATE using `rdrobust` and two MSE-optimal bandwidths. Interpret your results and compare it the model with a single optimal bandwidth.

Reveal Answer

```
robust_msetwo=rdrobust(educ$school_women, educ$margin, c=0, bwselect="msetwo")
summary(robust_msetwo)
```

This now looks pretty similar to the single optimal bandwidth, which is a good sign. In fact, the optimal bandwidth for data points above the cut-off remains virtually unchanged. For the ones below the cut-off, the bandwidths is slightly extended. The point estimate and measures of uncertainty also remain virtually unchanged. If specifying two optimal bandwidths alters the results significantly, this is an indicator that the data should be inspected closely for the cause of the diverging bandwidths.