Regression Discontinuity 2

Eduardo Fe

01/07/2019

Introduction.

The goal of this lab is to implement the local randomization method to continue the study of the effect of incumbency on electoral outcomes. To this end, we use another library, `rdlocrand`, written by Matias Cattaneo et al[1], as well as the data set provided by these authors, which contains data from elections to the US Senate. They focus is again on the effect of the Democratic party winning Senate seat on its vote share in the following election.

The local randomization approach relies on the existence of pre-treatment covariates in order to find a neighbourhood around the cutoff where treatment is allocated *as-if* at random. Their data includes the following variables to do this: state-level Democratic percentage of the vote in the pastpresidential election, state population, Democratic percentage of the vote in the t-1 Senate election,Democratic percentage of the vote in thet-2 Senate election, indicator for Democratic victory in thet-1Senate election, indicator for Democratic victory in thet-2 Senate election, indicator for open Senate seatatt, indicator for midterm (non-presidential) election attand indicator for whether the president of the U.S.

The first step, as usual is to install and load the library,

```
library(rdlocrand)
```

then, we load the data, which is already incorporated in the package,

```
data <- read.csv("D:/teaching/cattaneo/rdrobust_senate.csv")
```

We first need to find the bandwidth within which to undertake the analysis. To this end, we select the pre-treatment covariates mentioned above and saved them in a separate data set, say X

```
X  <- cbind(data$presdemvoteshlag1, data$population/1000000,
            data$demvoteshlag1,  data$demvoteshlag2, data$demwinprv1,
            data$demwinprv2, data$dopen, data$dmidterm, data$dpresdem)
```

To faciliate the interpretation of the results, we can assign more readable names to the variables,

```
R = data$demmv
Y = data$demvoteshfor2
D = as.numeric(R>=0)
```

Next, we compute and store the window width within which to undertake the analysis,

```
tmp = rdwinselect(R,X)
```

```
##
##
## Window selection for RD under local randomization
##
## Number of obs  =         1298
## Order of poly  =         0
## Kernel type    =         uniform
## Reps           =         1000
```

[1]Cattaneo, Frandsen and Titiunik (2015): Randomization Inference in the Regression Discontinuity Design: An Application to Party Advantages in the U.S. Senate, Journal of Causal Inference 3(1): 1-24.

```
## Testing method =         rdrandinf
## Balance test   =         diffmeans
##
## Cutoff c = 0             Left of c           Right of c
## Number of obs            600                 698
## 1st percentile           6                   7
## 5th percentile           30                  34
## 10th percentile          59                  70
## 20th percentile          120                 139
##
## Window length / 2       p-value      Var. name       Bin.test     Obs<c  Obs>=c
##
## 0.529                     0.189     X4                0.327        10     16
## 0.755                     0.271     X7                0.2          15     24
## 0.982                     0.065     X7                0.135        17     28
## 1.209                     0.101     X7                0.134        21     33
## 1.435                     0.134     X7                0.47         31     38
## 1.662                     0.113     X8                0.657        38     43
## 1.888                     0.044     X8                0.675        43     48
## 2.115                     0.079     X8                0.543        45     52
## 2.342                     0.123     X7                0.566        51     58
## 2.568                     0.15      X7                0.928        60     62
##
##
## Recommended window is [-0.755;0.755] with 39 observations (15 below, 24 above).
```

The output of `rdwinselect` is divided in three panels. The upper panel indicates the totalsample size, the degree of the polynomial and the type of kernel used for the weighting scheme (uniform, triangular, etc), the number of replications in the permutation test (whenever this test is performed), the method used to perform the covariate balance tests, the test statistic used.

The middle panel provides information on sample sizes. The first row gives the total number ofobservations to the left and to the right of the cutoff, and also the total sample size. The following four rows provide the same information but around small neighborhoods around the cutoffs defined by the first, fifth, tenth and twentieth percentile of the running variable. Finally, the main panel gives the result of the two balance tests performed at each of the windows considered.

Once the window has been selected, randomization inference to test the sharp null hypothesis of no treatment effect can be performed using `rdrandinf`.

```
tmp = rdrandinf(Y,R,wl=-.75,wr=.75)
```

Like `rdwinselect`, the output of `rdrandinf` is divided in three panels. The upper panel gives the total sample size, the order of the polynomial, the type of kernel used for the weighting scheme, the number of replications in the randomization test, and whether the window was specified by the user by setting or calculated using `rdwinselect`. The middle panel provides the number of observations at each sideof the cutoff, sample size below and above the cutoff inside the specified window, some descriptive statistics for the outcome inside the window, and the selected window. Note that the first line in this panel displays the number of observations with non-missing values of the outcome and running variable, so the sample sizes shown can differ from the total sample size. Finally, the main panel gives the results from the randomization test. The first column, labeled "Statistic", indicates the statistic used in the randomization test. The second column gives the observed value of the selected statistic and the third column shows its finite-sample p-value obtained from the randomization test. The fourth column gives the asymptotic p-value, that is, the p-val ueobtained from the corresponding asymptotic distribution of the chosen statistic. Finally the fifth column gives the asymptotic power against an alternative value that can be specified using the options `d()` or `dscale()`.

As mentioned above,`rdrandinf` uses the difference in means as the default statistic, but it can also use the

Kolmogorov-Smirnov and the rank sum statistics. By adding `statistic(all)` as an option we can obtain the result for all three statistics. If you try this you will see that the results of the three tests are essentially the same.

The `rdrandinf` command allows the user to specify a polynomial transformation model for the outcomes using the option `p`. By default, the command sets `p=0`. You can try now different values of `p` to see how your results might change; for instance, we could do

```
tmp = rdrandinf(Y,R,wl=-.75,wr=.75, p=1)
```

to take into account of any trends in the data within the window width. You can also modify the window width by setting up the lower and upper bounds with the options `wl` (window-left) and `wr` (window right).

The poor and the pious

Now that you have an understanding of the `rdrandifn` package, you can revisit Meyersson's data set and see how the results from your analysis vary when you change the estimation technique.