

Section 8: Testing for manipulation in a RD design - McCrary (2008)

Yotam Shem-Tov

Fall 2015

Detecting and testing for manipulation

- The best visual tool for detecting a manipulation around a specific point is *histogram* plots.
- The standard tool for hypothesis test is McCrary's Test following [McCrary \(2006\)](#).
- In this section we will cover McCrary (2008) with examples and code.

Notation

- Y_i is the outcome variable.
- $D_i \in \{0, 1\}$ is a binary treatment indicator.
- The relationship between the outcome and the treatment is,

$$Y_i = \alpha_i + \beta_i \cdot D_i = \bar{\alpha} + \bar{\beta} \cdot D_i + \epsilon_i$$

Is this different than potential outcomes framework?

No! $Y(0) = \alpha_i$ and $Y(1) - Y(0) = \beta_i$. The residual term is $\epsilon_i = \alpha_i - \bar{\alpha} + (\beta_i - \bar{\beta}) \cdot D_i$.

- The running variable is, R_i .
- Denote by R_{i0} the running variable if there was no intervention, no treatment.
- R_{i0} is an unobserved index. It is the counterfactual of R_i when there is no treatment, i.e., no incentive for individuals to manipulate R_i at the cut-point.
- There is manipulation in R_i if $R_i \neq R_{i0}$.

Assumptions

- The identification assumption is continuity with respect to the running variable.
- Let $f_{R_{i0}}(r)$ be the *pdf* of the running variable.
- The following three conditions are the identifying assumptions:

$$\mathbb{E}[\alpha_i | R_{i0} = r], \quad \mathbb{E}[\beta_i | R_{i0} = r] \quad \text{and} \quad f_{R_{i0}}(r)$$

are continuous in r .

Manipulation of R_i

- Does manipulation of the running variable necessarily lead to identification problems?
- Answer: *No!*
- When the manipulation is not associate (correlate) with any possible confounders, it is possible to meaningful parameter (causal relationships)
- Formally denote the possible confounders by W , there are no identification problems as long as, $f_{R|W}(r|w)$ is continuous
- Example: Randomly choose individuals who will be allowed to manipulate the running variable, R_i . In this context there is no correlation between the confounders, W , and the manipulation of the running variable.

Manipulation is not a sufficient condition for identification problems

The idea behind the test

- The idea behind the test is that continuity in $f_{R|W}(r|w)$ implies continuity in $f_R(r)$.
- This implies that we can use the observed density of the running variable to test for manipulation.
- Will the observed density always be informative for detecting manipulation? **No**
- When manipulation is not monotonic. Different individuals have an incentive to manipulate the running variable to different directions.
- Manipulation is monotonic if $R_i \geq R_{i0}$ is preferable for all i . Only in this case the observed density will be meaningful to detect manipulation of the forcing variable.

Hypothesis testing

- The McCrary test conducts the following hypothesis testing
- Assume the manipulation is *monotonic*, i.e $R_i \geq R_{i0}$ or $R_i \leq R_{i0}$
- The hypothesis are,

$$H_0 : R_i \leq R_{i0}$$

$$H_1 : R_i > R_{i0}$$

or

$$H_0 : R_i \geq R_{i0}$$

$$H_1 : R_i < R_{i0}$$

- When the manipulation is not monotonic it is possible to easily find examples in which the McCrary test will not detect manipulation (for example see the Article, pp 5)

Estimation

- The first step histogram:

$$g(R_i) = \lfloor \frac{R_i - c}{b} \rfloor \cdot b + \frac{b}{2} + c$$

- $g(R_i)$ belongs to the set,

$$\{\dots, c - 5 \cdot \frac{b}{2}, c - 3 \cdot \frac{b}{2}, c - \frac{b}{2}, c + \frac{b}{2}, c + 3 \cdot \frac{b}{2}, c + 5 \cdot \frac{b}{2}, \dots\}$$

- What does this set mean?
- This are the mid points in an histogram plot, in which the width of the bins is b

Estimation

- Implementation in R ,

```
b=0.8
```

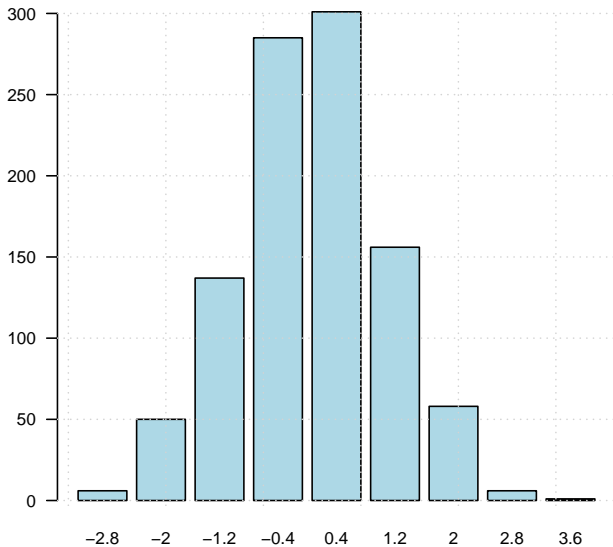
```
c=0
```

```
x=rnorm(1000,mean=0,sd=1)
```

```
g = floor((x-c)/b)*b+b/2+c
```

```
barplot(table(g),col="lightblue")
```

- The code plots an histogram in which the the points in g are the centres of the bars



Estimation

- Define an equal spaced grid X_1, X_2, \dots, X_J of width b covering the support of $g(R_i)$

- Define

$$Y_j = \frac{1}{n \cdot b} \cdot \sum_{i=1}^n 1(g(R_i) = X_j)$$

- Implementation in R,

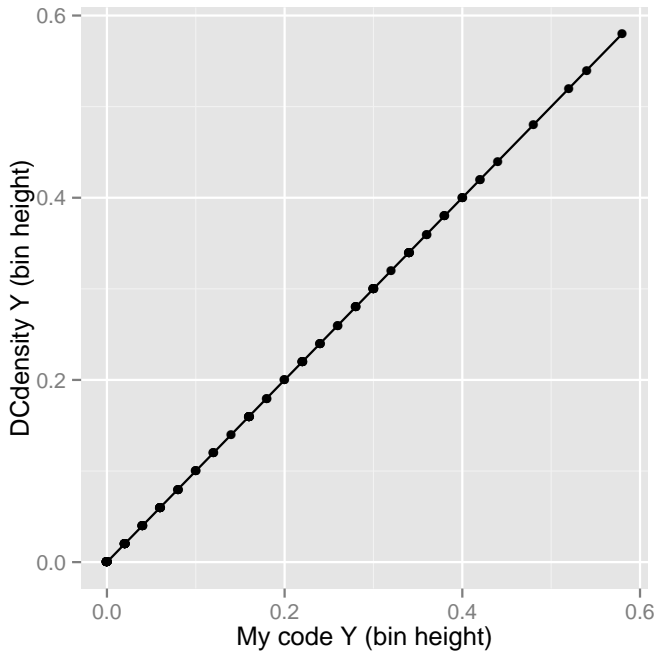
```
X = seq(min(g), max(g)+b, by=b)
```

```
Y = rep(999, length(X))
```

```
for (j in c(1:length(X))) {
```

```
  Y[j] = (1/(n*b))*sum(abs(g-X[j])<0.1^6)
```

```
}
```



Density estimation of $f(r)$

- In order to estimate $f(r)$ there are two steps:
 - ① Divide r into J bins
 - ② Fit a Kernel regression over the bins (not over the observations)
- In the previous slide we defined Y_j and X_j . The next step is to use them in order to estimate the density for each point r
- Fit a separate kernel regression procedure at each side of the cut-point, i.e one kernel regression for $r > c$ and another for $r < c$

Density estimation of $f(r)$

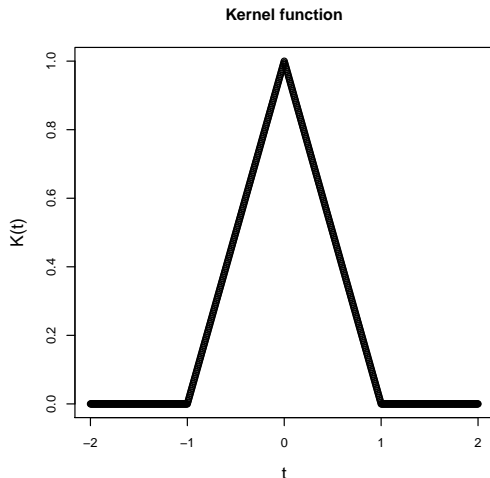
- For each value r , estimate a kernel regression over the bins (not the observations)
- The parameters $(\hat{\phi}_1, \hat{\phi}_2) = \operatorname{argmin} L(r, \phi_1, \phi_2)$, where,

$$L(r, \phi_1, \phi_2) = \sum_{j=1}^J \{Y_j - \phi_1 - \phi_2 \cdot (X_j - r)\}^2 \cdot K\left(\frac{X_j - r}{h}\right)$$

- This is exactly a weighted OLS regression in which the weight of observation j is, $\text{weight}_j = K\left(\frac{X_j - r}{h}\right)$
- The estimator for $f(r)$ is $\hat{f}(r) = \hat{\phi}_1$

The Kernel function

- What is $K(\cdot)$?
- $K(t) = \max\{0, 1 - |t|\}$



The Kernel function

- There are many different kinds of Kernel functions, which generates different weights. There is no limitation of using the *triangle* kernel
- Common Kernels are; "gaussian", "rectangular", "triangular", "epanechnikov", "biweight"
- As the bandwidth (h) is smaller there will be less observations with a positive weight, a non-zero kernel value.

The parameter of interest

- The parameter of interest is,

$$\theta = \ln \left(\lim_{r \downarrow c} f(r) \right) - \ln \left(\lim_{r \uparrow c} f(r) \right)$$

- The point estimate: see McCrary (2008) equation (4)
- In the paper McCrary proves (see the proposition) that $\hat{\theta}$ converges to a Normal distribution, and suggests an analytical formula for $\hat{\sigma}_\theta$

Simple example

- Consider the following data:

```
x<-runif(1000,-1,1)
```

```
x<-x+2*(runif(1000,-1,1)>0 & x<0)
```

- Write code that calculates the local linear regression from each side of the cut point and plot it in a figure.
- Use the binsize and bandwidth that are the default in the function “DCdensity”,

```
a=DCdensity(x,cutpoint=0,ext.out=TRUE,plot=FALSE)
```

```
h=a$bw
```

```
b=a$binsize
```

- The cut-point is zero, $c = 0$.

Solution: Step 1: binning

```
c=0
g = floor((x-c)/b)*b+b/2+c
X = seq(min(g),max(g)+b,by=b)

Y = rep(NA,length(X))
for (j in c(1:length(X))){
  Y[j] = (1/(n*b))*sum(abs(g-X[j])<0.1^6)
}
```

Solution: Step 2: writing a local linear function

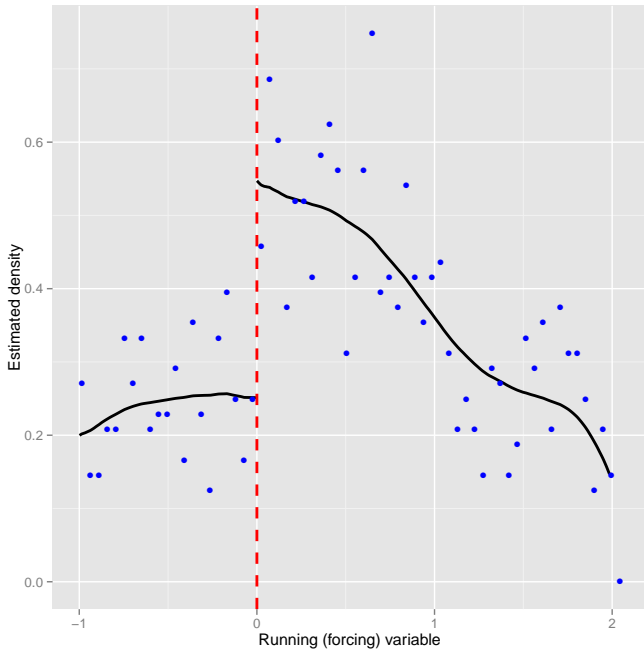
```
f.kernel.lm = function(r,side,bandwidth,X,Y,cutpoint){  
  
  K = apply(matrix((X-r)/bandwidth,ncol=1),1,function(x){return(max(0,1-abs(x)))})  
  X1 = (X-r)  
  
  # side (below or above "c"):  
  if (side=="left"){  
    K1 = K[X<cutpoint & r<cutpoint]  
    X1 = X1[X<cutpoint & r<cutpoint]  
    Y1 = Y[X<cutpoint & r<cutpoint]  
  }  
  if (side=="right"){  
    K1 = K[X>cutpoint & r>=cutpoint]  
    X1 = X1[X>cutpoint & r>=cutpoint]  
    Y1 = Y[X>cutpoint & r>=cutpoint]  
  }  
  
  lm.kernel = lm(Y1~X1+I(X1^2)+I(X1^3)+I(X1^4),weights=K1)  
  return(coef(lm.kernel)[1])  
}
```

Solution: Step 3: Estimation for all values of the running variable

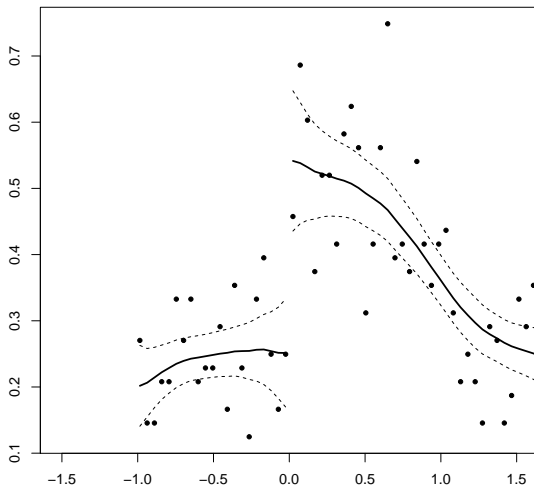
```
f.density1 = rep(NA,length(x))

max.x = length(x)
for (i in c(1:max.x)){
  cat("Iteration: ",i," out of ",max.x,"\n")
  if (x[i]<c){
    side0="left"
  }
  if (x[i]>=c){
    side0="right"
  }

  f.density1[i] = f.kernel.lm(x[i],side=side0,bandwidth=h,X=X,Y=Y,cutpoint=c)
}
```



The “DCdensity” default plot



Estimator of θ

- The estimator for θ is,

$$\hat{\theta} = \ln \left(\lim_{r \downarrow c} \hat{f}(r) \right) - \ln \left(\lim_{r \uparrow c} \hat{f}(r) \right)$$

- Write an **R** function that calculates $\hat{\theta}$ for a given cut-point, and verify your function yields the same answer as “DCdensity”.


```

f.estimate <- function(cutpoint,X,Y){
  r=cutpoint
  X.left = X[X<=cutpoint]
  Y.left = Y[X<=cutpoint]
  K.left = apply(matrix((X.left-r)/h,ncol=1),1,function(x){return(max(0,1-abs(x)))})
  X1.left = X.left-r
  lm.kernel.left = lm(Y.left~X1.left,weights=K.left)

  X.right = X[X>=cutpoint]
  Y.right = Y[X>=cutpoint]
  K.right = apply(matrix((X.right-r)/h,ncol=1),1,function(x){return(max(0,1-abs(x)))})
  X1.right = X.right-r
  lm.kernel.right = lm(Y.right~X1.right,weights=K.right)
  theta <- log(coef(lm.kernel.right)[1])-log(coef(lm.kernel.left)[1])
  return(theta)
}

theta.obs <- f.estimate(cutpoint=0,X=X,Y=Y)

# Check:
a=DCdensity(x,cutpoint=0,ext.out=TRUE,plot=FALSE)
a$theta

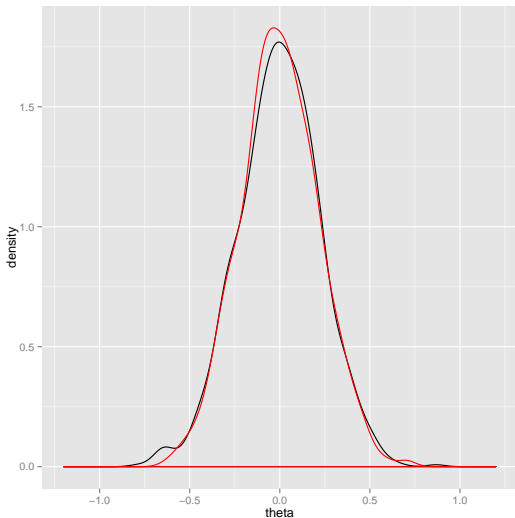
```

The SE of $\hat{\theta}$

- We can calculate $SE(\hat{\theta})$ using its limiting distribution or using a bootstrap procedure.
- The SE using Bootstrap is 0.235 and using the limiting distribution it is 0.206.

```
B=1000
boot.matrix <- matrix(sample(c(1:length(Y)),length(Y)*B,replace=TRUE),
                      ncol=B,
                      nrow=length(Y))
theta.boot <- rep(NA,B)
for (j in c(1:B)){
  index = boot.matrix[,j]
  Xb <- X[index]
  Yb <- Y[index]
  theta.boot[j] <- f.estimate(cutpoint=0,X=Xb,Y=Yb)
}
```

The limiting distribution based on $\widehat{SE}(\theta)$ and the bootstrap distribution of $\hat{\theta}$



The choice of *Binsize* and *Bandwidth*

- See the paper.

Application in *R*

- The package "rdd"
- The function *DCdensity* implements the McCrary density test,

```
DCdensity(running,cutpoint,ext.out=TRUE)
```