Section 6 : Regression Discontinuity II

Andrew Bertoli

9 October 2013

Roadmap

1. Equivalence Tests
2. QQ Plots
3. Bootstrapping
4. Regression Discontinuity Graphs
5. Questions

Equivalence Tests

Motivation

1. Standard practice is to draw a window around the cut-point and run balance tests or placebo tests for important covariates

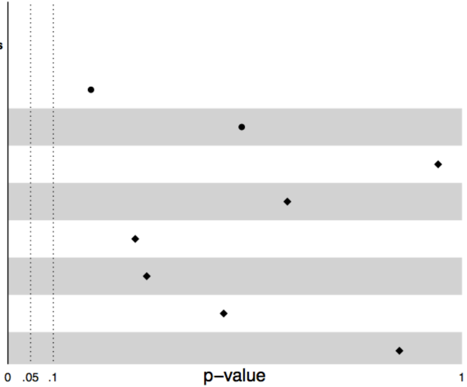
2. However, these tests assume that there is no difference between the two groups, giving an unfair advantage to the researcher.

3. Also, less data makes the study more likely to pass the balance or placebo tests, so the incentives to get more data are reversed.

4. In fact, if we got enough data, the balance or placebo tests would be significant for most designs, because the groups on either side of the cut-point will tend to be slightly different.

Equivalence Tests

Variable Name	Valid Cases	D Win Mean	D Loss Mean
Dem Sec of State	85	0.47	0.31
Dem Governor	85	0.40	0.48
Dem Pres % Margin	79	−0.09	−0.10
Voter Turnout %	85	37	34
Pct Gov't Worker	73	5.1	4.4
Pct Urban	73	70	65
Pct Black	73	4.9	5.0
Pct Foreign Born	73	4.0	4.1



Equivalence Tests

Basic Idea

1. Instead of using the null that the two groups are the same, equivalence tests start with the null that the two groups are dissimilar.

2. The alternative hypothesis is that the two groups are the similar, although they might not be exactly the same (proving that they are the same would be impossible).

3. A common way of defining similarity is to say that the means of the two groups are within 0.2 standard deviations of a covariate.

Equivalence Tests

R Code

1. Install the “equivalence” package.

2. The “tost” function is the standard difference in means equivalence test.

3. Plug in the treatment group for x, the control group for y, the test size for α (normally 0.05), and the length of region of similarity for ϵ (normally 0.2 times the standard deviation of the covariate).

4. If your data is paired, plug in the vector of differences between each pair for x and leave y blank.

5. If the null is rejected, then the two groups are considered similar

Equivalence Tests

R Code

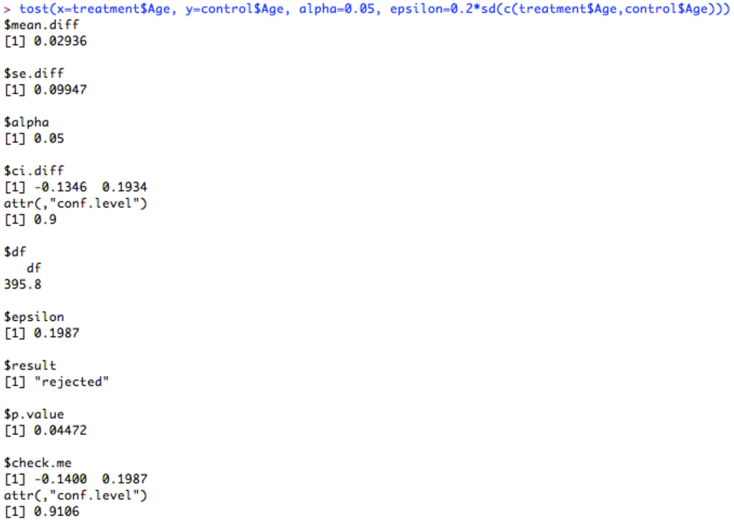
If our data is not paired

```
tost(x=treatment$Age, y=control$Age, alpha=0.05,
epsilon=sd(c(treatment$Age, control$Age)))
```

If our data is paired

```
differences=treatment$Age-control$Age

tost(x=differences, alpha=0.05, epsilon=0.2*sd(differences))
```

Equivalence Tests



```
> tost(x=treatment$Age, y=control$Age, alpha=0.05, epsilon=0.2*sd(c(treatment$Age,control$Age)))
$mean.diff
[1] 0.02936

$se.diff
[1] 0.09947

$alpha
[1] 0.05

$ci.diff
[1] -0.1346  0.1934
attr(,"conf.level")
[1] 0.9

$df
   df 
395.8 

$epsilon
[1] 0.1987

$result
[1] "rejected"

$p.value
[1] 0.04472

$check.me
[1] -0.1400  0.1987
attr(,"conf.level")
[1] 0.9106
```

Equivalence Tests

Problem

Makes the covariate tests different than the outcome test.

If you have a lot of data and no treatment effect, you might reject the null of a meaningful difference and reject the null of no treatment effect, simply because there is so much data.

With balance or placebo tests, we at least use the same procedures to evaluate the covariates as the outcome.

Conclusion: We should not abandon the normal balance an placebo tests.

Equivalence Tests

For more information on equivalence tests, see

Hartman and Hidalgo, “Whats the Alternative?: An Equivalence Approach to Balance and Placebo Tests.”

QQ Plots

A QQ plot (quantile-quantile plot) is a simple way to test for balance in the data.

Step 1: Calculate the quantile values for the two samples.

Step 2: Pair the values

(Min_1, Min_2),..., $(Median_1, Median_2)$,..., (Max_1, Max_2)

Step 3: Plot the pairs

QQ Plots

Example

Say we had the ages of two groups of 100 people each

Simply sort the ages into ascending order

The first pair is the youngest from each group, and the second pair is the second youngest from each group. You continue up until you pair the oldest of each group.

Then simply plot the pairs.

QQ Plots

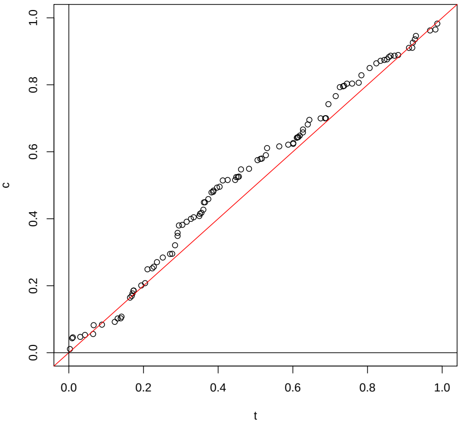
R Code

```
qqplot(t$Age,c$Age,xlim=c(-5,5),ylim=c(-5,5))

abline(0,1,col=”red”)

abline(v=0)

abline(h=0)
```

QQ Plots

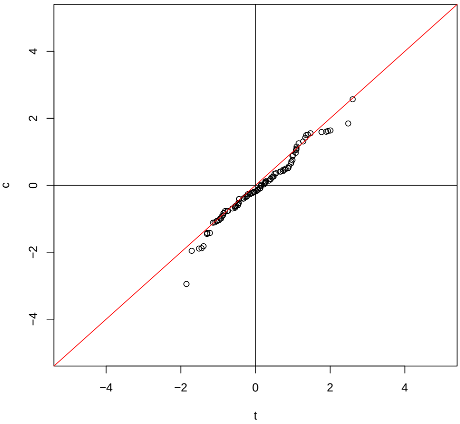
For each QQ plot, describe the relationship between the treatment and control groups.

QQ Plots



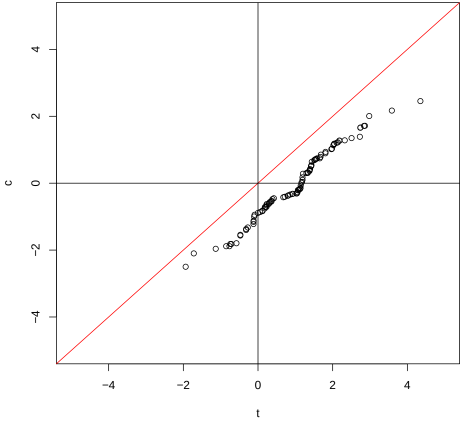
Same Distribution (uniform)

QQ Plots



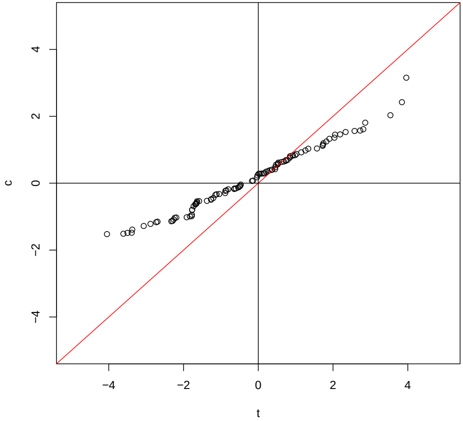
Same Distribution (normal)

QQ Plots



Treatment group has higher mean.

QQ Plots



Treatment group has higher variance.

QQ Plots

Advantage

Balance or equivalence tests only compare the means of the treatment and control group, whereas QQ plots compare the entire distributions.

Disadvantages

Need a graph for every covariate.

Less easy to interpret.

Bootstrapping

Motivation

Imagine that we want to study the height of people in the United States, so we take a random sample of 10,000 Americans and measure them.

By the Central Limit Theorem, the sample average $\hat{\mu}$ will be distributed about N(μ, $\sigma/100$), where μ is the population mean and σ is the sample variance.

However, what if we are interested in the population median? The CLT does not give us a simple way to estimate its distribution.

Bootstrapping

Basic Idea

To estimate the distribution of a parameter like the sample median

1. Take many random samples (with replacement) from the original sample, each of size n (in this case 10,000)

2. Estimate the parameter for each sample

3. The distribution of these estimates should be close to the sampling distribution of the estimator

Bootstrapping

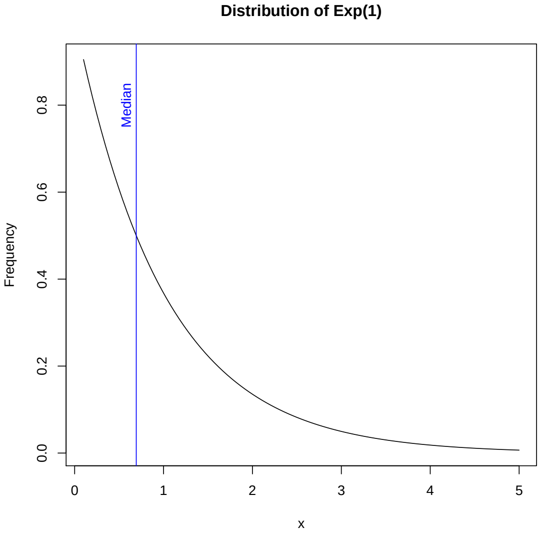
Example

Say we want to find the median of an Exponential(1) distribution. To get our estimator, we will take a sample of 100 draws from this distribution, and then take median of these draws. What is the sampling distribution of this estimator?

To estimate the sampling distribution, we will take 10,000 samples with replacement from our original sample of 100. The size of each new sample will be 100. We will then take the median of each sample.

The distribution of these medians will be close to the sampling distribution of our estimator.

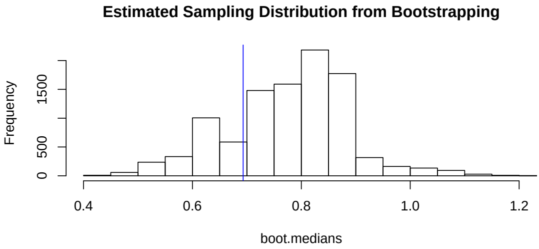
Bootstrapping

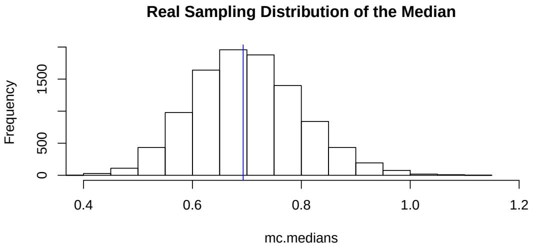


Bootstrapping

```
original.sample=rexp(n=100, rate=1)

boot.medians=rep(0,10000)

for(i in 1:10000){

new.sample=sample(original.sample, size=100, replace=TRUE)
boot.medians[i]=median(new.sample)

}

mean(boot.medians)

hist(boot.medians, main=”Estimated Sampling Distribution from Bootstrapping”)
```

Bootstrapping





Bootstrapping

Hypothesis Tests and Confidence Intervals

1. Imagine that we our interested in some parameter θ, and we take B bootstrapped samples to estimate θ

2. Our estimator is $\bar{\theta}^* = \frac{1}{B}\sum_{i=1}^{B} \hat{\theta}_i^*$

3. Our estimated standard error is

$$\hat{se}_B = \sqrt{\sum_{i=1}^{B}(\theta_i^* - \bar{\theta}^*)/(B-1)}$$

4. The estimated 95% confidence interval is

$$\bar{\theta}^* \pm 2 * \hat{se}_B$$

5. If we are doing the normal hypothesis test, we reject the null if the estimate is more than 2 standard deviations away from the 0 (meaning that 0 is outside the 95% confidence interval).

Bootstrapping

Percentile Bootstrap

1. You can also estimate the 95% more using the empirical quantiles
2. Start by ordering the $\hat{\theta}_i^*$ from lowest to highest
3. The 95% confidence interval is $(\hat{\theta}^*_{(2.5)}; \hat{\theta}^*_{(97.5)})$

Bootstrapping

Regression Example

1. Since our regression model is $Y_i = X_i\beta + \epsilon_i$, where the ϵ_i are i.i.d., we can start by sampling with replacement from the e_i's.

2. After drawing many random samples, we will have a large number of vectors of bootstrapped residuals, which we will call the $\mathbf{e}^*$'s

3. For each of the $\mathbf{e}^*$'s, we can estimate a vector $\mathbf{Y}^*$ using the formula $\mathbf{Y}^* = \mathbf{X_i}\boldsymbol{\beta} + \mathbf{e}^*$.

4. We can estimate the $\hat{\boldsymbol{\beta}}^{\boldsymbol{*}}$'s by using the formula $\hat{\boldsymbol{\beta}}^{\boldsymbol{*}} = [\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{Y}^*$

5. If we are interested in the parameter β_i, the distribution of the $\hat{\beta}_i^* - \hat{\beta}_i$ should be a good approximation of $\hat{\beta}_i - \beta_i$

Regression Discontinuity Graphs

Constructing the Local Linear Regression Lines

1. Use the ksmooth() function in R to construct lines on both sides of the cut-point.

```
ksmooth(x=forcing.variable,y=outcome,
bandwidth=opt.bandwidth)
```

2. Get the optimal bandwidth using Devin Caughey's rdoptband_catch function

```
rdoptband_catch(forcingVar=forcing variable for entire sample,
outcomeVar=outcome for entire sample sample,
cutpoint=cut-point)
```

3. You can use these same steps to construct regression lines for covariates.

Regression Discontinuity Graphs

Estimating the Confidence Interval by Bootstrapping

1. Starting with the left side of the cut-point, calculate the residuals for each observation.

2. Randomly sample (with replacement) from these residuals. The sample size should be the same as the number of points on the left. Add these residuals to the predicted values of the outcome.

3. Construct a regression line for this new sample

4. Repeat this process several thousand times. This will result in thousands of regression lines. Each regression line will be defined by a large number of (x, y) coordinates. Thus, each x will be associated with a large number of y's.

5. For each x, find the lower 2.5% and upper 97.5% quantile for the y's. These points will define the lower and upper bounds of the 95% confidence interval.

6. Repeat these steps to get the confidence interval for the right.