Section 8 : Local linear regression (loess) with an application to RDD

Yotam Shem-Tov

Fall 2014

Motivation

- We will focus on local linear regression (henceforth *loess*) in the context of RDD
- The purpose of using *loess* is to estimate:
    - The continuity of the covariates (pre-treatment variables) with respect to the running variable
    - Estimating (and visualizing) the effect of the treatment (the running variable) on the outcome of interest
    - Kernel smoothers and *loess* can be used to test for manipulation of the running variable around the cut-point
- **The loess is used in order to estimate a continuous function of a variable of interest with respect to the running variable.**

The RDD model

- Let Y_i be the observed value of a variable of interest (either the outcome or a covariate)
- Let $(Y_i(0), Y_i(1))$ be the pair of potential outcomes of unit i, and Let $W_i \in \{0, 1\}$ be a treatment indicator, and X_i is the running (forcing) variable
- We consider only a sharp RDD in which, $W_i = 1(X_i > 0)$
- Define the average treatment effect on Y at $X = x$ as,

$$\tau(x) = \mathbb{E}(Y_i(1) - Y_i(0)|X_i = x)$$

- Regression discontinuity methods focus on estimating the average effect of the treatment at the threshold $(X = 0)$,

$$\tau = \tau(0) \Leftrightarrow \tau = \mathbb{E}(Y_i(1) - Y_i(0)|X_i = 0)$$

The RDD model

- Under smoothness of the conditional expectations of the potential outcomes as a function of the forcing variable, τ can be estimated as the discontinuity in the conditional expectation of Y_i:

$$\begin{aligned}\tau &= \underbrace{\lim_{x\downarrow 0} \mathbb{E}(Y_i(1)|X_i = x)}_{\mu_+} - \underbrace{\lim_{x\uparrow 0} \mathbb{E}(Y_i(0)|X_i = x)}_{\mu_-} \\ &= \underbrace{\lim_{x\downarrow 0} \mathbb{E}(Y_i|X_i = x, X_i \geq 0)}_{\mu_+} - \underbrace{\lim_{x\uparrow 0} \mathbb{E}(Y_i|X_i = x, X_i \leq 0)}_{\mu_-}\end{aligned}$$

- Our objective is to estimate μ_- and μ_+ under the assumption the functions $\mathbb{E}(Y_i(1)|X_i = x)$ and $\mathbb{E}(Y_i(0)|X_i = x)$ are smooth,i.e. continuous and differentiable at $X = 0$
- What are $\hat{\mu}_-$ and $\hat{\mu}_+$?

The RDD model

- Estimate $\mathbb{E}(Y_i|X_i = x)$ and $\mathbb{E}(Y_i|X_i = x)$ using a smoother function.
- Denote:

$$f_0(x) = \widehat{\mathbb{E}}(Y_i|X_i = x, X_i \leq 0)$$
$$f_1(x) = \widehat{\mathbb{E}}(Y_i|X_i = x, X_i \geq 0)$$

Hence,

$$\hat{\tau} = \underbrace{f_1(x=0)}_{\hat{\mu}_-} - \underbrace{f_0(x=0)}_{\hat{\mu}_+}$$

- What is our objective when choosing a smoother? Is our objective to estimate $\mathbb{E}(Y_i|X_i = x, X_i \leq 0)$ and $\mathbb{E}(Y_i|X_i = x, X_i \geq 0)$ most accurately? *No! Our objective is to estimate μ_- and μ_+ most accurately. We do not need to estimate most accurately points which are far from $x = 0$, i.e. we want the most accurate estimation at the cut-point*

The RDD model

- Why do we want to estimate the conditional expectation of Y_i using a "smoother"? *Because we assume the conditional expectation function is "smooth"!*
- We will discuss 3 different methods:
  1. Global high order polynomial regression (*Parametric estimation*)
  2. Local linear regression (LOESS) (*Non-parametric estimation*)
  3. Nadaraya-Watson Kernel regression estimator (*Non-parametric estimation*)
- Global high order polynomial regression: We estimate $\mathbb{E}(Y_i|X_i = x, X_i \leq 0)$ by fitting the regression model,

$$X_i \leq 0, \quad Y_i = \sum_{j=0}^{K} X_i^K \beta_{-j} + \epsilon_i \Rightarrow f_0(x) = \sum_{j=0}^{K} X_i^K \hat{\beta}_{-j}$$

where K is the degree of the high order polynomial regression

Application: Lee (2008)

- Lee (2008) used close elections to estimate the incumbency advantage
- Caughey and Sekhon (2011) show there is evidence of sorting around the cut-point, and imbalance in the observed covariates
- We will demonstrate how to fit a smoother to the data using two data sets:
  - (i) The replication data from Caughey and Sekhon (2011) - Polynomial regression smoother and a local linear regression smoother
  - (ii) Lee's original data - Polynomial regression smoother
- If the RDD identifying assumptions hold, there should not be a discontinuity of the pre-treatment variables (covariates) with respect to the running variables at the cut-point
- We will test this testable implication with respect to the variable **DifDPPrv**, Democratic margin of victory in the previous election

Lee (2008): Polynomial regression

- A popular approach is to use a polynomial regression as a smoother
- A polynomial regression of degree p is fitted to the data.
- In order to allow for a different polynomial fit in each side of the cut-point we interact all the polynomial terms (including the intercept) with an indicator variable for the treatment, indicator the observation is to the right of the cut-point
- The following regression model is fitted to the data,

$$
\begin{aligned}
Y_i &= \alpha + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_p X_i^p + \\
&= \gamma d_i + \delta_1 d_i X_i + \delta_2 d_i X_i^2 + \cdots + \delta_p d_i X_i^p
\end{aligned}
$$

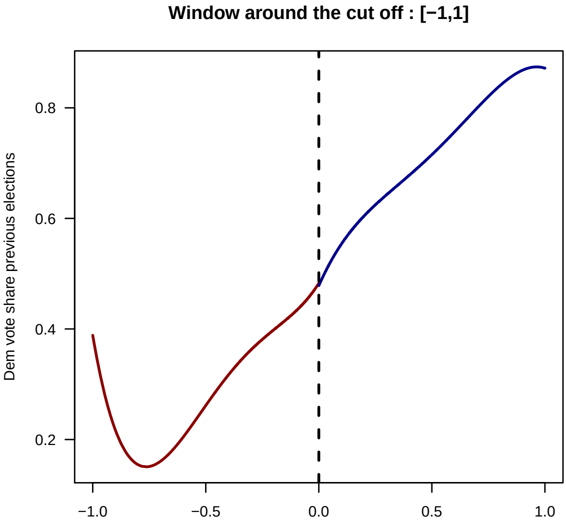
Y_i - *The variable of interest (DifDPPrv)*
X_i - *The running variable*
$d_i = 1(r_i > 0)$ - *An indicator variable whether observation i is to the right of the cut-point (received treatment)*

Lee (2008): Polynomial regression

- The parameter of interest is γ. What is the interpretation of γ?
  The intercept of the polynomial regression in the treatment group, i.e. whether there is a non-continuous jump in the polynomial fit at the cut-point
- The standard errors of γ are calculated under the assumption of a linear regression
- We will consider several different scenarios:
  1. Polynomial regression of degree 4, with Lee's original data (Not Caughey and Sekhon replication data), over 4 different window sizes, *window* $\in \{0.1, 0.25, 0.5, 1\}$
  2. Polynomial regression of degree, *degree* $\in \{2, 3, 4, 5\}$, using Caughey and Sekhon replication data, over a window size of 0.1 same as the one used in their paper

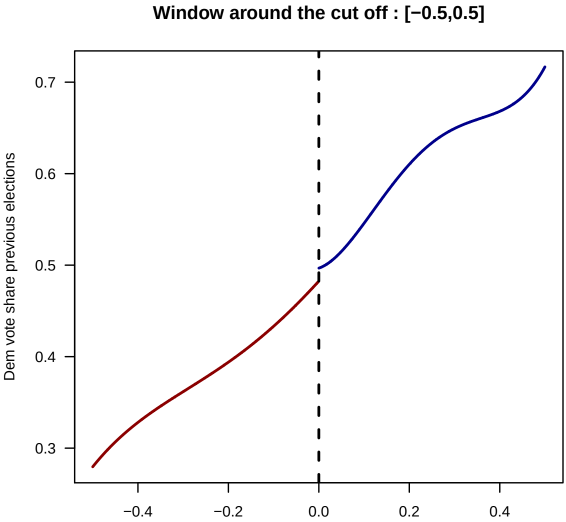
Lee (2008): Polynomial fit, scenario 1

Original Lee (2008) data using 4th order polynomial and different windows



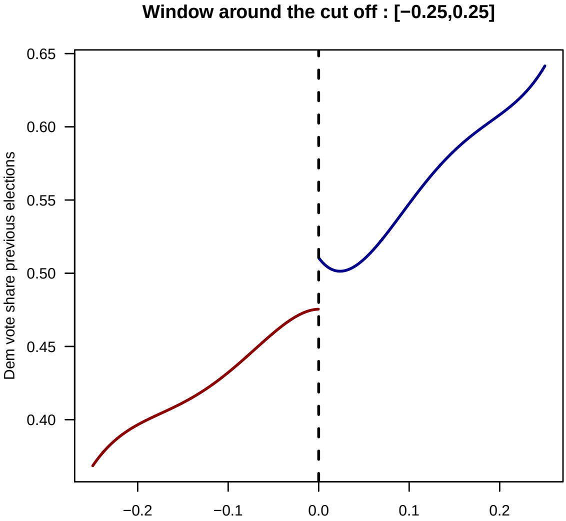
Lee (2008): Polynomial fit, scenario 1

Original Lee (2008) data using 4th order polynomial and different windows



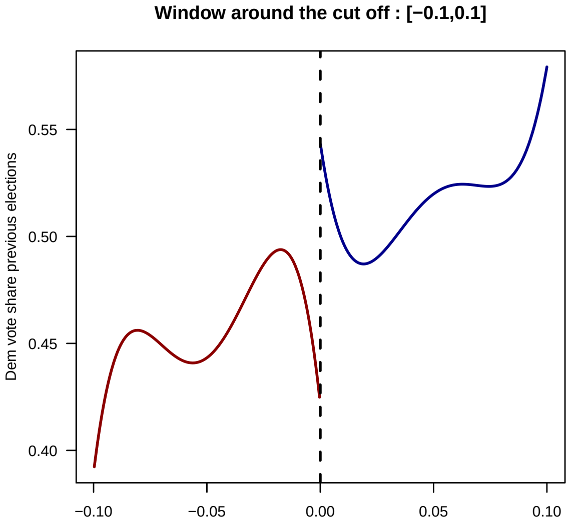
Lee (2008): Polynomial fit, scenario 1

Original Lee (2008) data using 4th order polynomial and different windows



Lee (2008): Polynomial fit, scenario 1

Original Lee (2008) data using 4th order polynomial and different windows

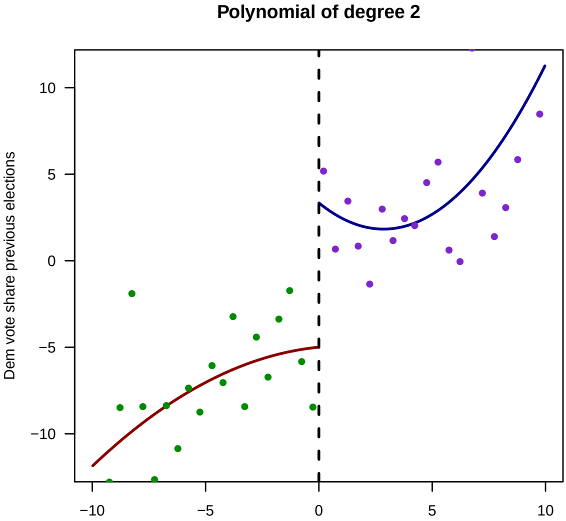


Application: Lee (2008)

- Is the previous vote share (pretreatment variable) continuous with respect to the running variable? at the cut-point
- Next we use Caughey and Sekhon (2011) replication data to fit different polynomial regression of different degrees
- The next step will be to consider *loess* instead of polynomial regression

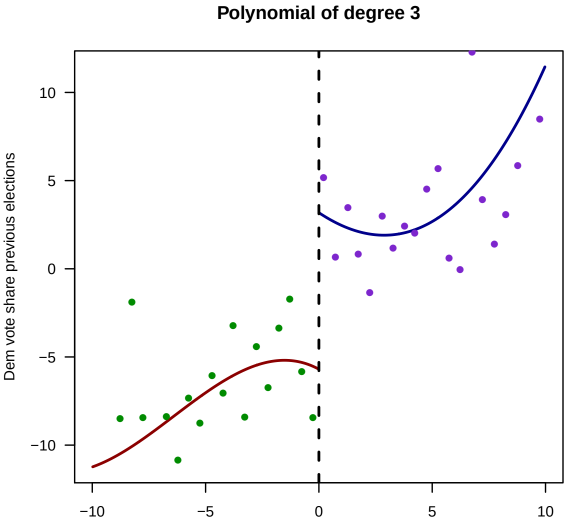
Lee (2008): Polynomial fit, scenario 2

Caughey and Sekhon (2011) data using different polynomial degrees



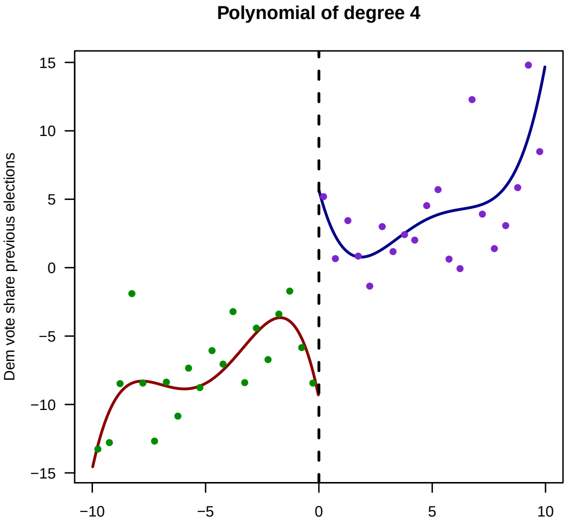
Lee (2008): Polynomial fit, scenario 2

Caughey and Sekhon (2011) data using different polynomial degrees



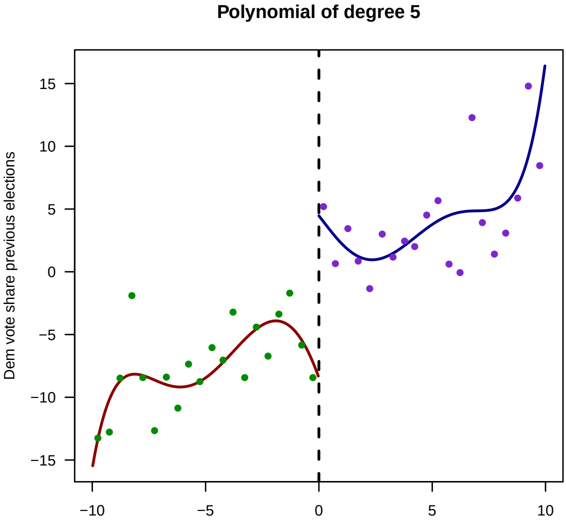
Lee (2008): Polynomial fit, scenario 2

Caughey and Sekhon (2011) data using different polynomial degrees



Lee (2008): Polynomial fit, scenario 2

Caughey and Sekhon (2011) data using different polynomial degrees



Lee (2008): Hypothesis testing, scenario 2

Caughey and Sekhon (2011) data using different polynomial degrees

What is the coefficient for which we want to perform hypothesis testing? The coefficient which represents the cut-point, γ

Polynomial fit using Caughey and Sekhon (2011) replication data

	Polynomials 2	Polynomials 3	Polynomials 4	Polynomials 5
(Intercept)	-4.99	-5.70	-9.50^{*}	-8.44
	(2.60)	(3.51)	(4.42)	(5.32)
right	8.34^{*}	8.88	15.18^{*}	12.90
	(3.77)	(4.96)	(6.12)	(7.25)
R^2	0.06	0.06	0.06	0.06
Adj. R^2	0.06	0.05	0.05	0.05
Num. obs.	1635	1635	1635	1635

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. Note, window size of 10 points (0.1)

code for reproducing the figure

```
### Bins: ###
n.bins=40
breaks0 = seq(-10,10,by=0.5)
bins0 = cut(d$DifDPct,breaks=breaks0)
bins = tapply(d$DifDPct,bins0,mean)
bin.DifDPPrv = tapply(d$DifDPPrv,bins0,function(x){return(mean(x
### Figure: ###
xx = lm2$model$DifDPct[order(lm2$model$DifDPct)]
yy = lm2$fitt[order(lm2$model$DifDPct)]

par(cex=0.7)
plot(xx[xx<0],yy[xx<0],las=1,col="red4",xlab="Runing variable (D
     ylab="Dem vote share previous elections",type="l",
     xlim=c(min(xx),max(xx)),ylim=c(min(yy),max(yy)),
     lwd=2,main="Polynomial of degree 2")
lines(xx[xx>0],yy[xx>0],col="blue4",lwd=2)
abline(v=0,lwd=2,col=1,lty=2)
points(bins,bin.DifDPPrv,ty\item The theoretical research on opt
col=ifelse(bins<=0,"green4","purple3"))
```

Global higher order polynomial regression

Gelman and Imbens (2014)

- Would we want to use global polynomial regression in practice? **No**
- Imbens and Gelman (2014) argue against the use of global higher order polynomial regression as a "smoother" in the RDD context.
- Imbens and Gelman discuss the following issues against the use of global higher order polynomial regression:
  1. **The weighted average representation of polynomial regressions**.
  2. **Estimates that are highly sensitive to the degree of the polynomial**
  3. **Inferences that do not achieve nominal coverage**

Global higher order polynomial regression

Gelman and Imbens (2014)

1.
   - An estimate based on a polynomial regression, with or without trimming, can be interpreted as the difference between a weighted average of the outcomes for the treated and a weighted average for the controls.
   - Given the choice of estimator, the weights depend only on the threshold and the values of the forcing variable, not on the values for the outcomes. One can, and should in applications, inspect these weights.
   - Gelman and Imbens argue the weights implied by polynomial regression have unattractive properties.

What does it mean "trimming"? Different window sizes around the cut-point

Can *loess* be represented as a difference in two weighted means? *Yes! However Gelman and Imbens argue that it has much better weights.*

Global higher order polynomial regression

Gelman and Imbens (2014)

2. Results based on high order polynomial regressions are sensitive to the order of the polynomial. Moreover, we do not have good methods for choosing the order in a way that is optimal for the objective of a good estimator for the causal effect of interest.

   Is it a good procedure to choose the degree of the polynomial by cross-validation? *It is better than guessing or choosing the degree without any justification, however this is not related to the research objective of causal inference.*
   Why not? *We are interested in good inference at the cut-point, not in the goodness of the polynomial approximation far from the cut-point. Hence, any global measure of accuracy will not be ideal*

Global higher order polynomial regression

Gelman and Imbens (2014)

3.
- Inference based on high-order polynomials is often poor. Specifically, confidence intervals based on such regressions, taking them as accurate approximations to the regression function, are often misleading.
- Even if there is no discontinuity in the regression function, high-order polynomial regressions often lead to confidence intervals that fail to include zero.
- There is a problem in the **Type I error**. Gelman and Imbens show an example for such a scenario.

LOESS

- Loess is equivalent to estimating many weighted OLS regressions around each observation
- The idea behind loess is to estimate each point (observation) separately,
  (i) Choose an interval around each observation, x_i (can be of different lengths)
  (ii) Fit a separate linear regression in each interval
- The intervals are not disjoint, they overlap each other.
- Loess "tuning" parameter is the size of the bandwidth (or bandwidths)
- How should we choose the bandwidth? To minimize the variance in the point of interest

LOESS: The model

- The data consist of n pairs of observations, $(x_1, y_1), \ldots (x_n, y_n)$
- The DGP (data generating process) is assumed to be,

  $$y_i = \mu(x_i) + \epsilon_i$$

  where $\mu(x)$ is an unknown continuous and differentiable function and $\epsilon_i \sim i.i.d$ with $\mathbb{E}(\epsilon_i) = 0$, is an error term
- For fitting a point x define a bandwidth, $h(x)$. The bandwidth will determine the size of the interval around x in which the observations has a positive weight when estimating what is $\mu(x)$
- The observations are weighted according to:

  $$w_i(x) = K\left(\frac{x_i - x}{h(x)}\right)$$

  where $w_i(x)$ is the weight of observation x_i when estimating $\mu(x)$, and $K(t)$ is a weight function that assigns largest weights to observations close to x, i.e. the Kernel function

LOESS: Estimation of $\mu(x)$

- $\mu(x)$ is approximated by a weighted polynomial regression of degree p
- $\mu(x)$ is estimated at the point $X_i = x$ by the regression equation,

$$\begin{aligned}\hat{\phi}_1, \ldots, \hat{\phi}_p = \underset{\phi_1,\ldots,\phi_p}{argmin} \sum_{i=1}^{n} K\left(\frac{x_i - x}{h(x)}\right) \cdot \{y_i - \phi_0 - \phi_1 \cdot (x_i - x) \\ - \phi_2 \frac{1}{2} \cdot (x_i - x)^2 - \cdots - \phi_p \frac{1}{p} \cdot (x_i - x)^p\}^2\end{aligned}$$

- When $p = 1$:

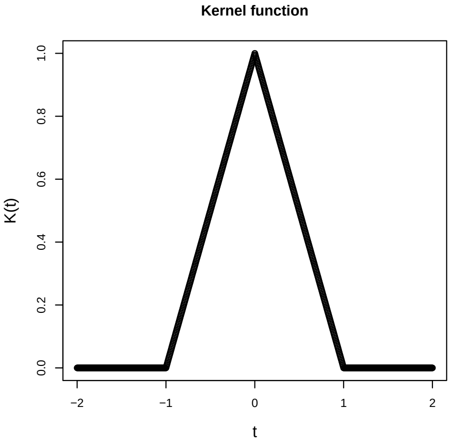
$$\hat{\phi}_0, \hat{\phi}_1 = \underset{\phi_1,\phi_0}{argmin} \sum_{i=1}^{n} K\left(\frac{x_i - x}{h(x)}\right) \cdot \{y_i - \phi_0 - \phi_1 \cdot (x_i - u)\}^2$$

- The estimator of $\mu(x)$ is, $\widehat{\mu}(x) = \hat{\phi}_0$

The Kernel function

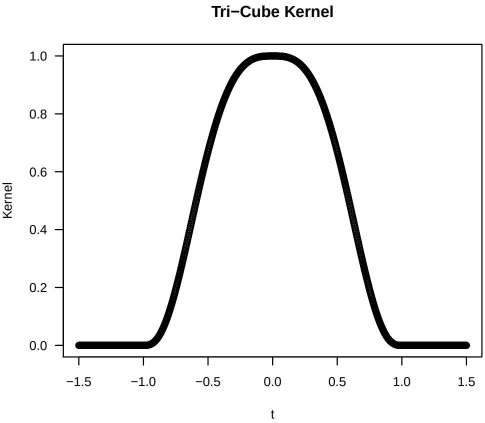
- There are many different kinds of Kernel functions, which generates different weights.
- Common Kernels are; "gaussian", "rectangular", "triangular", "epanechnikov", "biweight"
- McCrary (2006) uses the *triangle* kernel.
- In the popular R *locfit* function the default Kernel is, *Tri-cube*
- As the bandwidth (h) is smaller there will be less observations with a positive weight, a non-zero kernel value.

The Kernel function



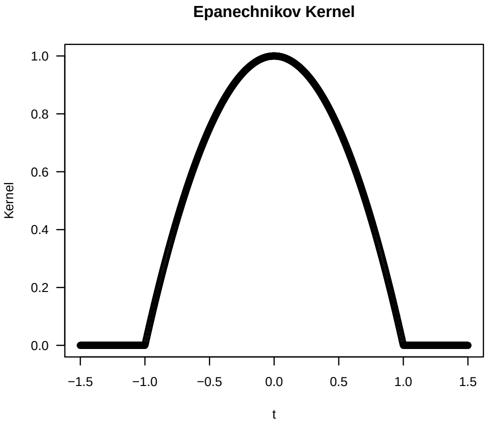
What is $K(t)$? $K(t) = max\{0, 1 - |t|\}$

The Kernel function



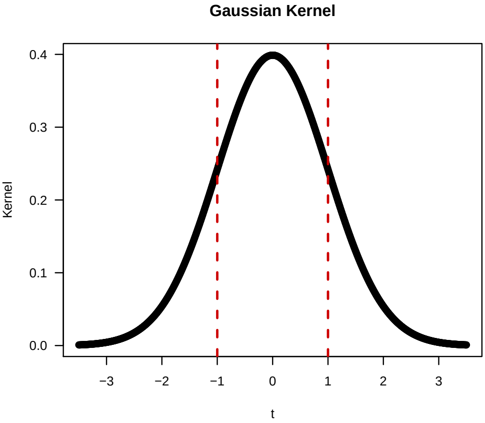
What is $K(t)$? $K(t) = \begin{cases} (1-|t|^3)^3, & \text{if} \quad |t| < 1 \\ 0, & \text{else} \end{cases}$

The Kernel function



What is $K(t)$? $K(t) = \begin{cases} (1 - t^2), & \textit{if} \quad |t| < 1 \\ 0, & \textit{else} \end{cases}$

The Kernel function



What is $K(t)$? $\frac{1}{\sqrt{(2*pi)}} \cdot exp\left(-\frac{t^2}{2}\right)$

Kernel functions: Examples of applications

Applications of Kernel functions:

1. Kernel regression, i.e. a kernel smoother
2. Density estimation: Kernel functions are used when estimating the density of a random variable
3. Caliper matching or caliper IPSW
4. In different kinds of "smoother" functions, such as *loess*
5. Text analysis and classification, text data mining

LOESS estimation: Bandwidths

- What are the "tuning" parameters in the LOESS procedure?
  1. The choice of the bandwidths, $h(x)$
  2. The polynomial degree in each local linear regression
- The difficult question is the choice of bandwidths, $h(x)$. There are two common simplifications of $h(x)$:
  - (i) $h(x) = h, \quad \forall_x$. The bandwidth is constant for all x
  - (ii) $\frac{|\{i:i \in h(x_i)\}|}{N}$ is constant. The fraction of the data which is included in each bandwidth is constant, for example 0.01, 0.2, 0.5.

    In *R* this is the parameter *span* in the function *loess* and in the *ggplot2* package when using the smoothing method *loess*
- The theoretical research on optimal bandwidth selection in the context of RDD focussed on the case of constant bandwidth, $h(x) = h, \quad \forall_x$

LOESS estimation: Bandwidths

- Two recent papers discuss data driven procedures for finding the optimal bandwidths:
  1. Calonicoy, Cattaneoz and Titiunik (2014) (henceforth CCT) in Econometrica
  2. Imbens and Kalyanaraman (2012) (henceforth IK) in Review of Economic Studies.
- IK assume a constant bandwidth, $h(x) = h$, and choose h^* by,

$$\begin{aligned} h^* &= \underset{h}{argmin}\; MSE(\hat{\tau}, \tau) \\ &= \underset{h}{argmin}\; \mathbb{E}\left[(\hat{\tau} - \tau)^2\right] = \mathbb{E}\left[(\hat{\mu}_+ - \hat{\mu}_- - (\mu_+ - \mu_-))^2\right] \\ &= \underset{h}{argmin}\; \mathbb{E}\left[(\hat{\mu}_+ - \mu_+ - (\hat{\mu}_- - \mu_-))^2\right] \end{aligned}$$

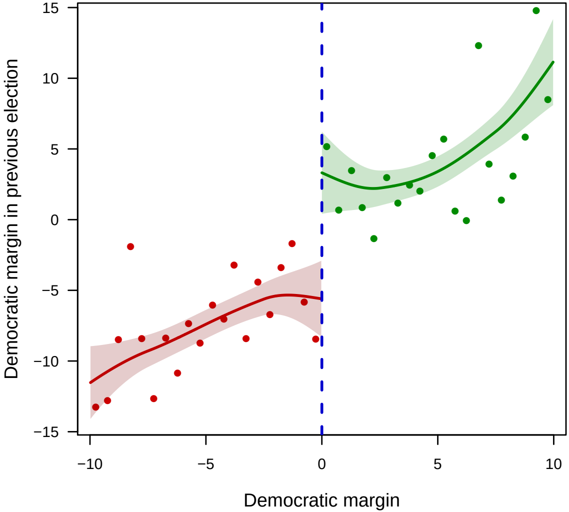
  where $\hat{\tau} = g(h)$, i.e $\hat{\tau}$ is a function of h
- h^* is chosen to minimizes the asymptotic approximation of $MSE(\hat{\tau}, \tau)$, denoted as $AMSE(\hat{\tau}, \tau)$

LOESS estimation: Bandwidths

- CCT argue the optimal bandwidth by IK is too large and suggest a different procedure.
- CCT suggest a double "robust" bandwidth selection procedure
- Next we show how to fit a LOESS smoother using Caughey and Sekhon (2011) replication data

- Implementation of RDD in *R*:
  1. The *rdd* package. It implements IK bandwidth selection procedure, McCrary test, and estimates treatment effect at the cut-point with "robust" standard errors
  2. The *rdrobust* package written by CCT. It implements the CCT bandwidths selection procedure. The package provides tools for data-driven graphical and analytical statistical inference in RD

LOESS estimation: Caughey and Sekhon (2011) replication data



LOESS estimation: The *locfit* function

- The *loess* was fitted using the function *locfit* in the package *locfit*. See also the function *locfit.raw* which shows all the parameters the user has control over and all the options in *locfit*.

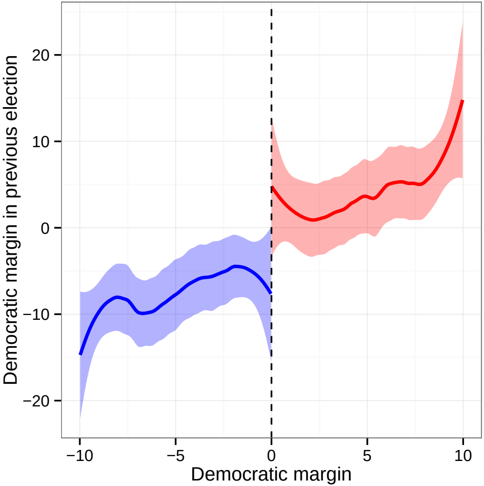
  ```
  # fitted "loess" of the control observations,
  fit0 <- locfit(DifDPPrv~DifDPct,data=d[d$DifDPct<0,],
  alpha=c(0.1^5,bandwidth),deg=1,kern="tcub")
  loess0.fit = predict(fit0,newdata=d[d$DifDPct<0,],
  se.fit=TRUE)
  ```

- The *locfit* function allows to fit a *loess*, with a fixed bandwidth or/and a *span* parameter. the parameter "alpha" has two inputs, the first is the *span* and the second is the fixed bandwidth. It will use the larger of the two. To remove the *span* parameter from the estimation set it to a small number, 0.1^5
- The choice of bandwidth is by IK using the package *rdd*

LOESS estimation: Caughey and Sekhon (2011) replication data

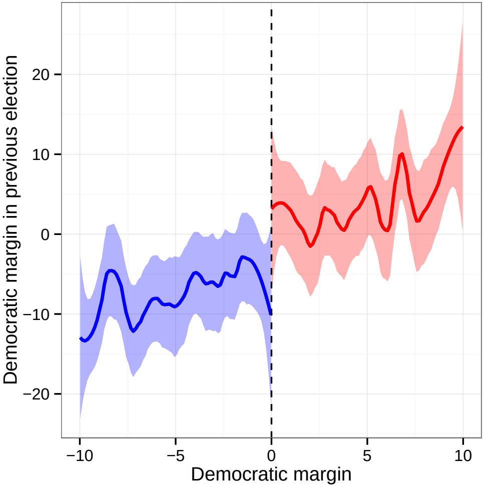
- How did we calculate the confidence intervals? *By taking one SE above and below the fitted smoother line. The locfit function calculates a SE for each estimated point, e.g every fitted observation has a different SE*
- The SE are calculated automatically using locfit and I conjecture using analytical formulas of the SE, which implies this are estimations of the asymptotic SE
- Can we use the *ggplot2* package and the *loess* built in smoother? *This can be problematic.*
- The *ggplot2* built in *loess* smoother uses the *span* parameter. As far as I know it does not allow for a constant bandwidth. Now we face the question of how to choose the *span* parameter? Both CCT and IK do not discuss the choice of *span*, but the choice of h

LOESS estimation: The *ggplot2* built in *loess* function



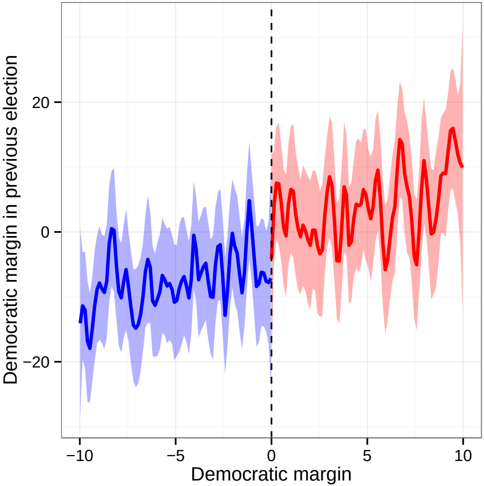
What is the *span* parameter? (Hint: $N = 895$) *span=0.5*

LOESS estimation: The *ggplot2* built in *loess* function



What is the *span* parameter? *span=0.25*

LOESS estimation: The *ggplot2* built in *loess* function

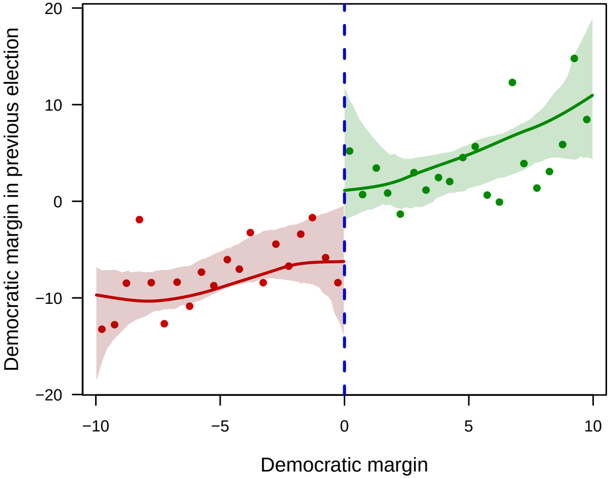


What is the *span* parameter? *span=0.1*

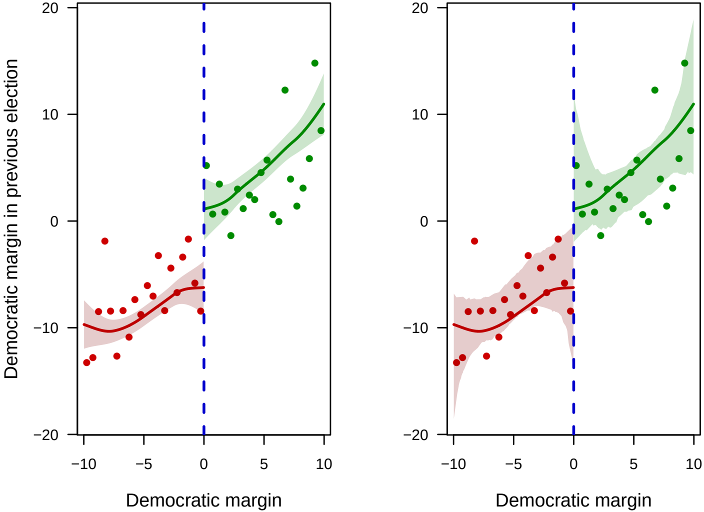
LOESS estimation: Calculating confidence intervals

- The SE can also be calculated using bootstrap
- Is there any special problem implementing bootstrap in this setting? *In each bootstrap sample the points are different. How can we calculate a confidence interval for point x?*
- Solution: We can fit a *loess* in each bootstrap sample and predict the fitted values at the points $(x_1, \ldots, x_N)$ in the original sample
- Should we choose the same bandwidth in all the bootstrap samples, i.e. fix the bandwidth over bootstrap samples? *No. We want to include also the variance which results from the choice of bandwidth. Our bandwidth selection procedure is data driven and hence must be recalculated in each bootstrap sample*

LOESS estimation: Bootstrap SE, *Replication=200*



LOESS estimation: Bootstrap SE vs Asymptotic SE

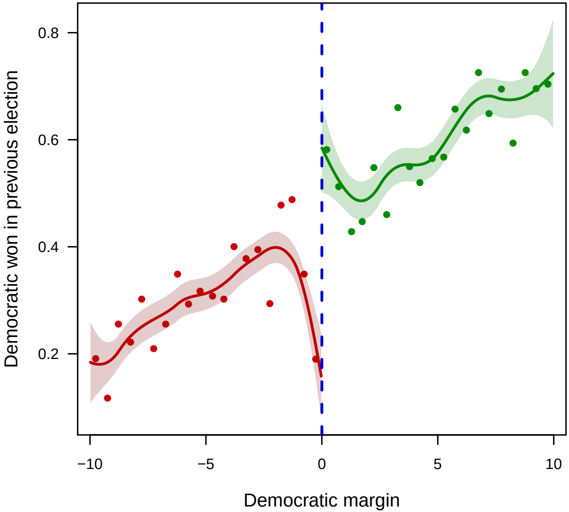


LOESS estimation: Binary outcome of interest

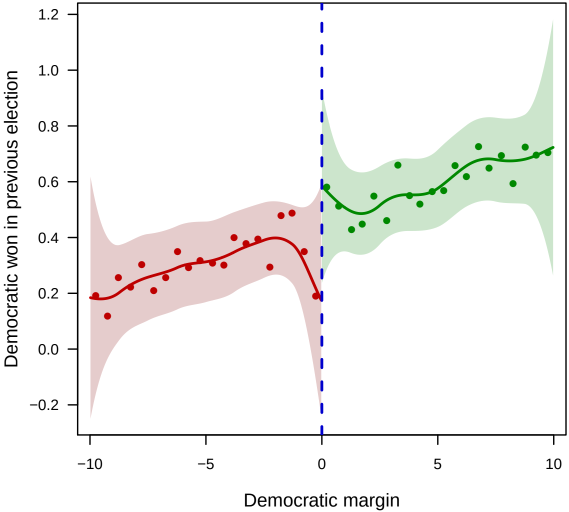
- We are interested whether there is a discontinuity at the cut-point with respect to a binary variable. Indicator variable whether a democrat won in the previous election, ”DWinPrv”
- We have two option: 1. Linear probability model (default in *loess*). 2. GLM: For example Logit.
- We can fit a local Logistic regression using *loess* by setting: family=”binomial”,link=”logit”

```
fit0 <- locfit(DWinPrv~DifDPct,data=d[d$DifDPct<0,],
   alpha=c(0.1^5,bandwidth),
    deg=1,kern="tcub",family="binomial",link="logit")
```

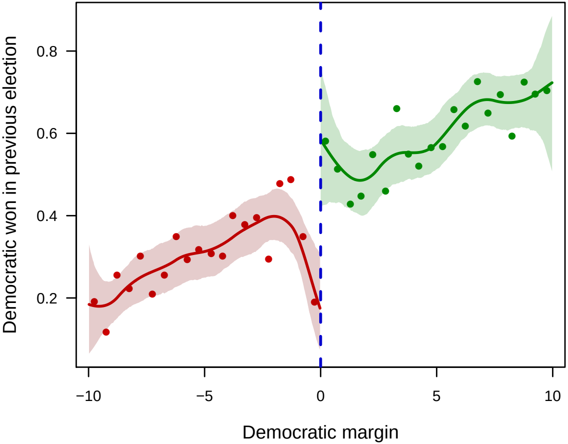
LOESS: Binary dependent variable, liner probability model



LOESS: Binary dependent variable, logistic regression



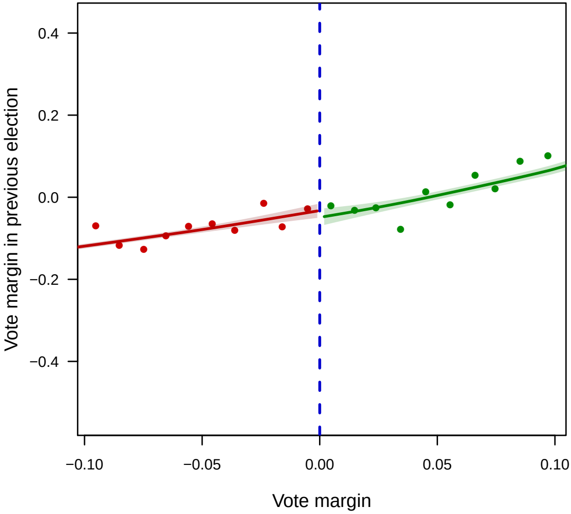
LOESS: Binary dependent variable, logistic regression, SE with bootstrap with 600 repetitions



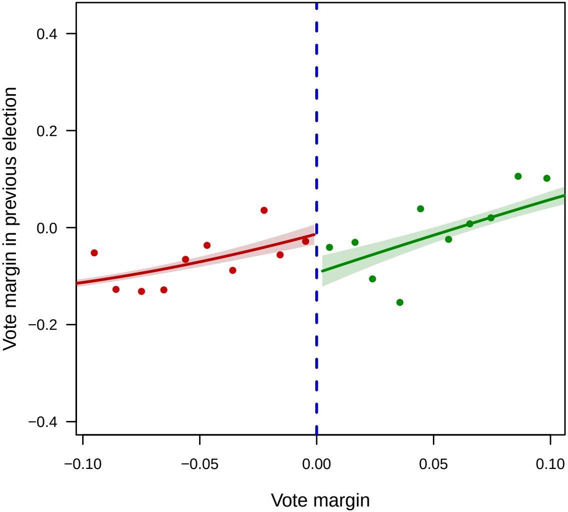
Example: MPs for Sale? Returns to Office in Postwar British Politics

- Next we look on the paper MPs for Sale? Returns to Office in Postwar British Politics by Eggers and Haimueller (2009)
- There are two options for inference in RDD settings:
  1. Assuming continuity of the potential outcomes with respect to the running variable and fitting a "smoother" function (LOESS curves)
  2. Assuming the observations in a window around the cut-point where "randomly" assigned to treatment, and analyse the data as we would an experiment. In this case is there testable implications? Yes, covariates balance and no-manipulation around the cut-point
- In the next figures we look on the testable implications of the RDD identification strategy in this data
- Note, all the LOESS models where fitted using all the data, however the results are presented for a window of 0.1 around the cut-point

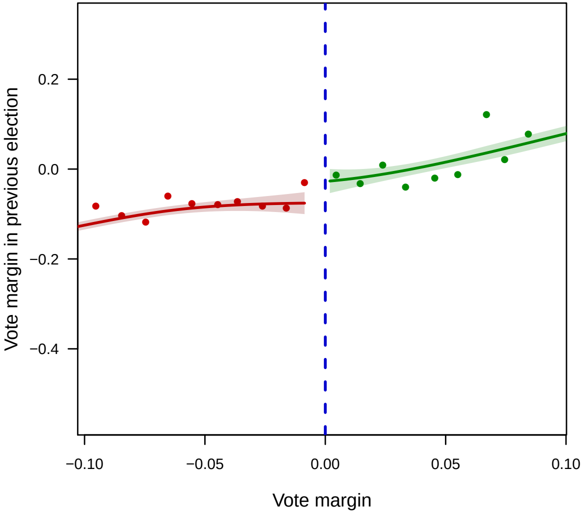
LOESS: MPs for Sale? Previous vote share all parties



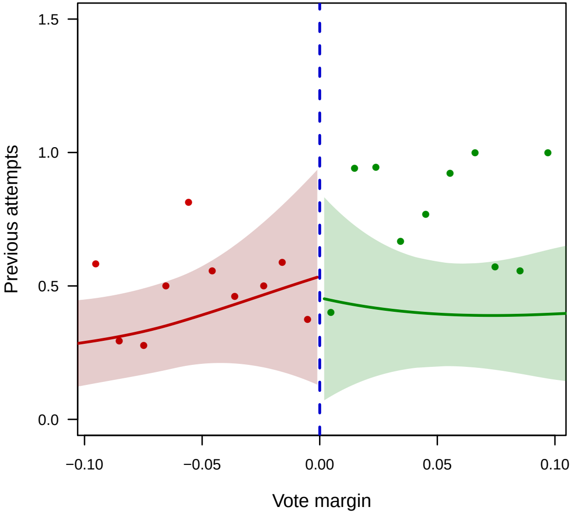
LOESS: MPs for Sale? Previous vote share only labour party



LOESS: MPs for Sale? Previous vote share only non-labour parties

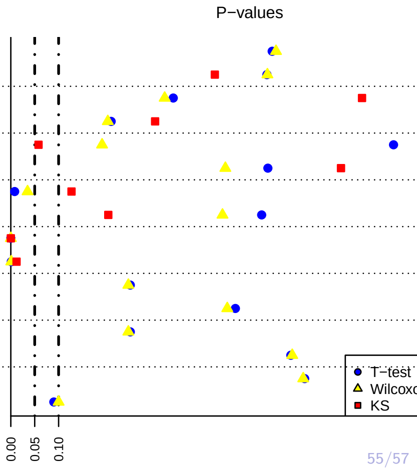


LOESS: MPs for Sale? Number previous of attempts



LOESS: MPs for Sale? Window of 0.1

	Mean Treated	Mean Control
xxfemale	0.036	0.052
xxyob	1917.609	1918.353
xxyod	1995.655	1994.866
age	78.045	76.481
wobl_race_effective_number_of_ca	2.248	2.239
wobl_race_electorate	58084.055	59062.903
prev_attempts	0.773	0.493
wobl_race_turnout	0.805	0.801
pre_wobl	−0.002	−0.081
labour	0.355	0.56
xxoc_local_politics	0.273	0.209
xxoc_civil_serv	0.018	0.007
xxoc_local_politics.1	0.273	0.209
xxoc_business	0.118	0.142
xxoc_white_collar	0.109	0.09
xxoc_journalist	0.064	0.127



Window around the cut-point

- How to choose the size of the window around the cut-point?
- An excellent paper, Cattaneo, Frandsen and Titiunik (2014)

References for farther reading and *R* implementation

- An excellent book for implementing LOESS in R, and the theory behind it in a practical approach is, "Local Regression and Likelihood" by Clive Loader
- The *locpol* package, has a variaty of useful functions and implementations. It implements the Fan and Gijbels(1996)'s Rule of thumb for bandwidth selection
- Fan and Gijbels(1996) book on local polynomial modelling and its applications.
- The GAM package is also useful to check out. It is based on the book "Generalized Additive Models" by Hastie and Tibshirani (1990)
- reading on the *rdrobust* package and examples of it's implementation