Advanced Topics in Applied Regression

Day 4: Nonparametric specifications

Constantin Manuel Bosancianu

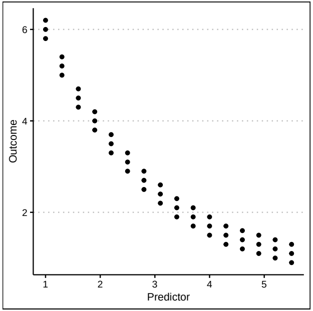
Doctoral School of Political Science
Central European University, Budapest
bosancianu@icloud.com

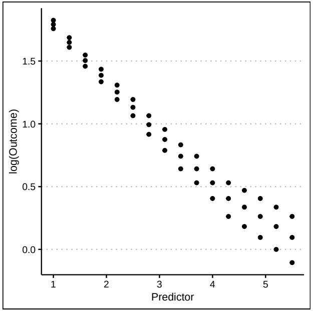
August 3, 2017

Non-parametric specifications

Earlier in the class we ran a regression of infant mortality on GDP/capita.

We addressed the difficulties with a transformation of GDP/capita.





Why go nonparametric . . .

Linearity of specification is essentially the default for most empirical testing.

Not very much thought is given to the possibility of nonlinear relationships $\Rightarrow$ we don't really have developed theories about functional forms in the social sciences.

Linear in parameters: $Y_i = a + b_1 X1_i + b_2 X2_i + e_i$.

Linear in parameters, but not in variables:
$Y_i = a + b_1 X1_i + b_2 X2_i + b_3 X2_i^2 + e_i$.

…when you have power transformations?

$$Turnout_i = a + b_1 \times Age_i + b_2 \times Age_i^2 + e_i \qquad (1)$$

However, a power transformation will change the shape of the relationship *globally*, not only in a specific section.

It's also the case that, usually, the choice of which power transformation to use is arbitrary: X^2, X^3, $\sqrt{X}$, …

Choosing based on a model fit criterion is not guaranteed to result in the proper model being selected.

Smoothers

Advantages:

- ✓ faster, and easier to present, than neural networks, support vector machines, or tree-based methods;
- ✓ still rely on the linear regression machinery;
- ✓ functional form of the model is not imposed on the data, but estimated from it.

However, they are considerably more computationally intensive than OLS. Additionally, they don’t produce tables of results, but a graphical representation.

Local Polynomial Regression

Local polynomial regression (LPR)

A better strategy is to model directly the process which generated the data points.

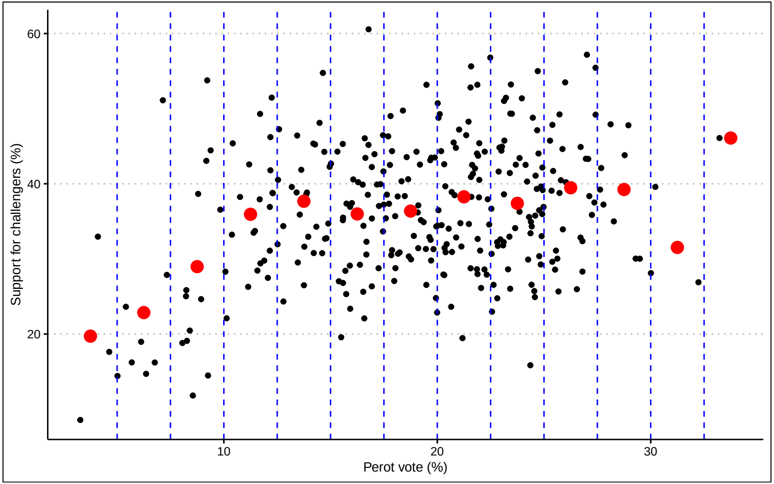
$$Y_i = f(X1, X2, \dots) + e_i \tag{2}$$

This $f(X1, X2, \dots)$ could be a standard linear specification, but also a nonlinear one estimated directly from the data.

All that the LPR expects is that the function be smooth.[1]

[1] In "math-speak", that the first-order derivative is defined at every point of the function.

Moving average smoother



Support for Perot in 1992 and vote for challengers (US House)

Constructing the bins

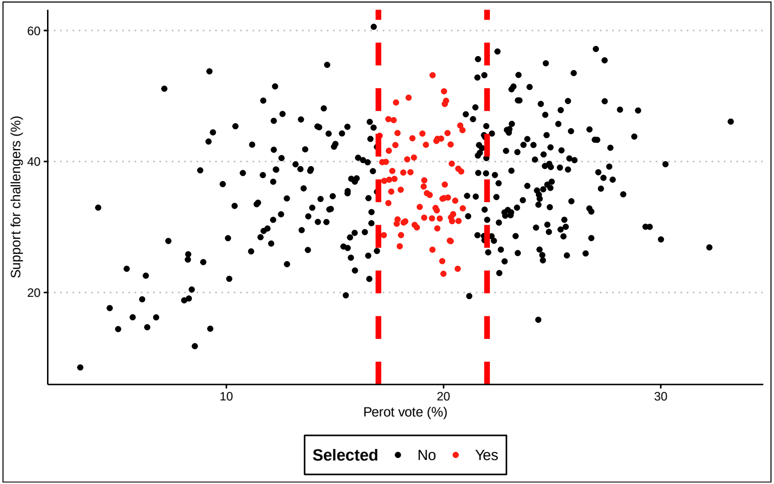
A lot rides on how to construct the bins: too narrow and variability of means increases, too wide and the trend appears too smooth.

A few strategies:

- ✓ bins of equal range (like above) – however, some might contain little data;
- ✓ bins with equal amounts of data;
- ✓ window bin which moves across X.

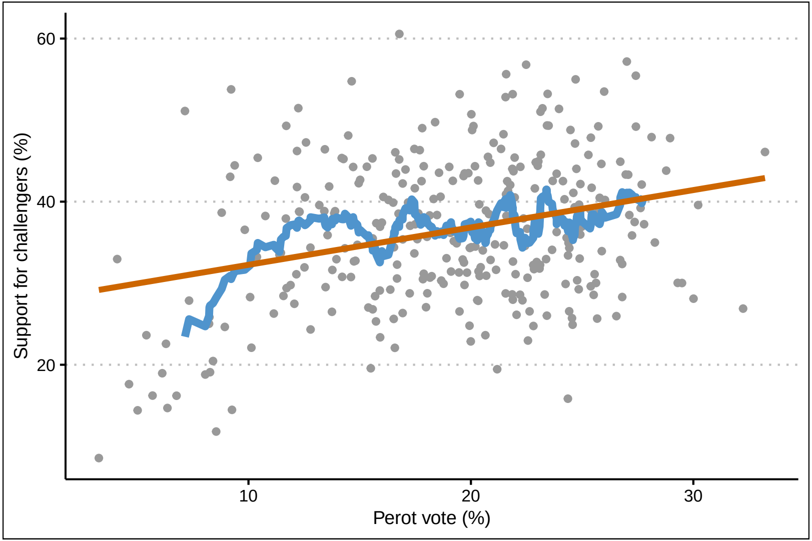
The last is most frequently used—observations move in and out of the window, and are used in computing the average.

Moving window process



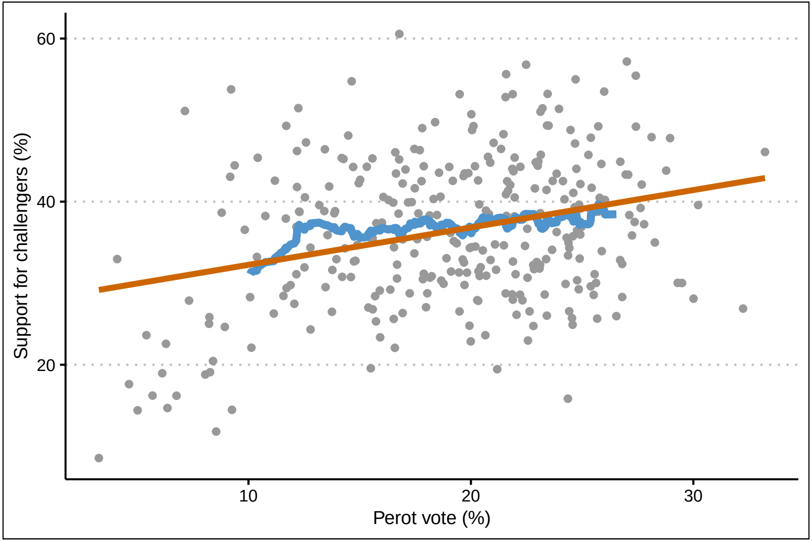
Moving window approach

Window width



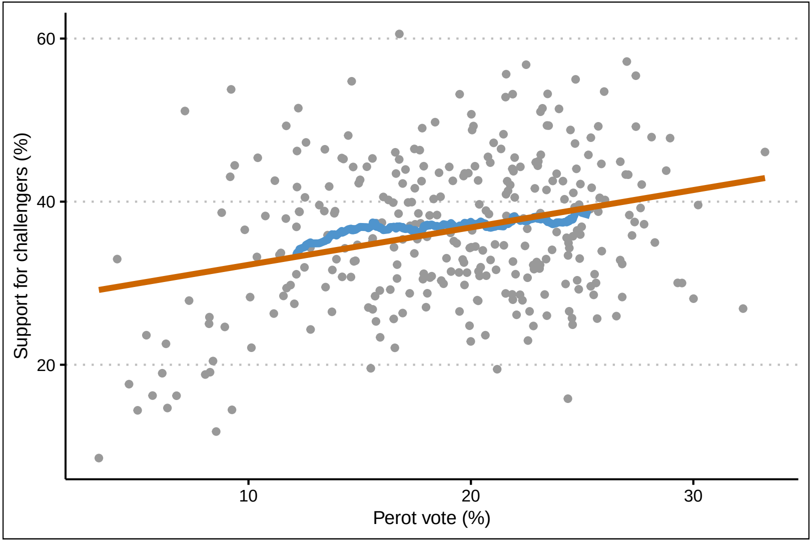
Effect of changing window width: 21

Window width



Effect of changing window width: 51

Window width



Effect of changing window width: 81

Kernel smoothing

One problem with the moving average smoother is that it allocates equal weight to all cases, irrespective of how far they are to the focal point (the center of the moving window).

Kernel smoothing addresses this by adding 2 extra steps to the mix:

- ✓ a "distance" measure from the center of the window;
- ✓ a weighting function, based on distance.

The average now becomes a weighted one, but nothing else changes.

Kernel smoothing

Distance measure: $z_i = \frac{x_i - x_0}{h}$

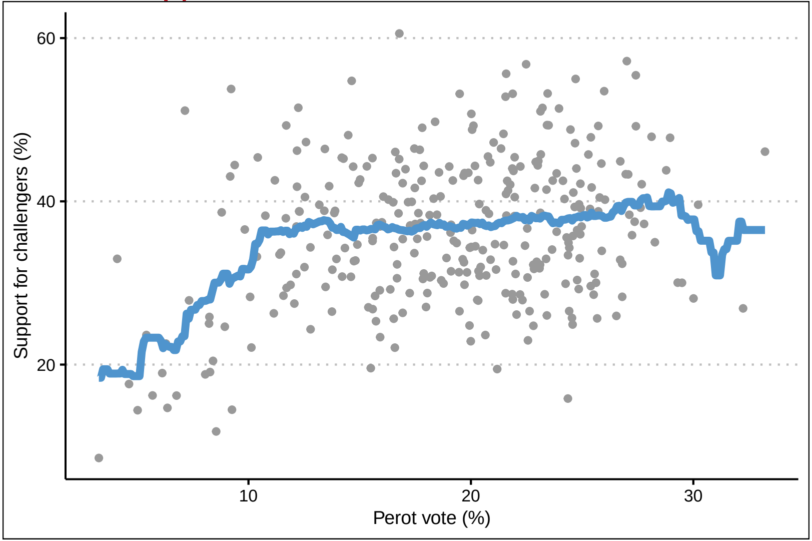
x_0 is the center of the window, and h is its width.

The most popular weighting function is the *tricube kernel*:

$$K_T(z) = \begin{cases} (1 - |z|^3)^3, & \text{for } |z| < 1 \\ 0, & \text{for } |z| \geq 1. \end{cases} \tag{3}$$

You can imagine the moving average as a weighted procedure with equal weights.

Kernel smoothing



Kernel smoothing with bin width of 4

LPR: benefits

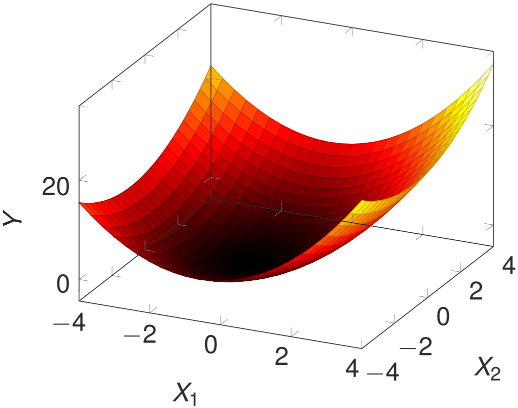
Both the moving average and the kernel smoother are essentially computing averages.

However, we can go beyond this and actually run a regression of Y on X. The two most famous procedures are *loess* and *lowess* (Cleveland, 1979).

$$Y_i = a + b_1 X_i + b_2 X_i^2 + \cdots + b_p X_i^p + e_i \tag{4}$$

In empirical work it's very rare to see $p > 3$.

Polynomial specifications



Equation: $Y = 2.5X_1 + 1.75X_2 + 2X_1^2 + X_2^2$

LPR: implementation (I)

A window width is chosen, usually in terms of % of data (similar to kernel smoothers).

Within each bin, a WLS estimation of the polynomial specification is conducted.

$$\frac{Y_i}{w_i} = \frac{a}{w_i} + b_1 \frac{X_i}{w_i} + b_2 \frac{X_i^2}{w_i} + \cdots + b_p \frac{X_i^p}{w_i} + \frac{e_i}{w_i} \tag{5}$$

The w_i are typically assigned with the tricube kernel used in kernel smoothing.

LPR: implementation (II)

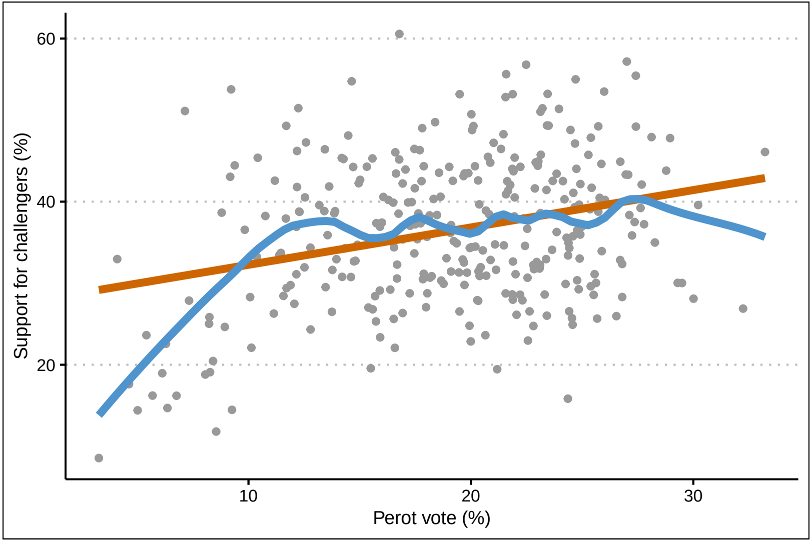
In a second stage, a set of robustness weights is obtained from the specification in Equation 5. These are then applied to the model, for another round of estimation.

Then we start again with the w_i, then with robustness weights.

The process stops when there is minimal change in estimates from one iteration to another.

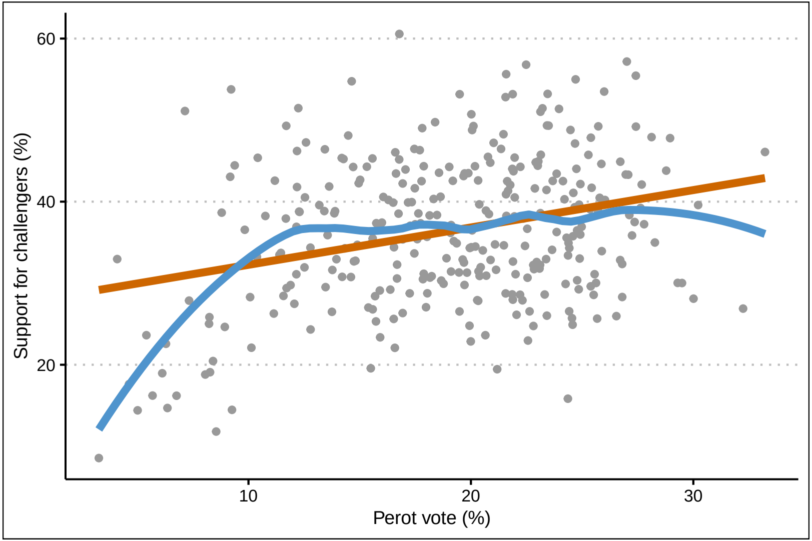
The use of w_i is what distinguishes *lowess* from *loess*.

LPR for Perot vote



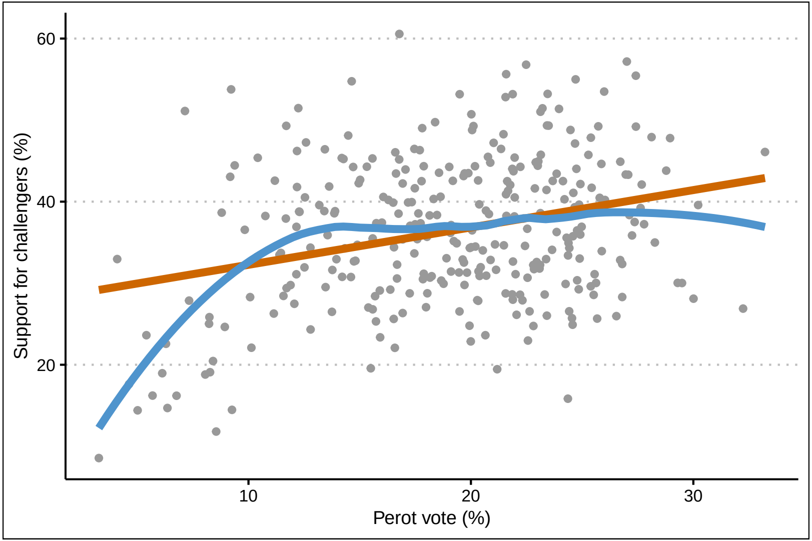
LPR with span of 0.30

LPR for Perot vote



LPR with span of 0.45

LPR for Perot vote



LPR with span of 0.60

Local polynomials: choices

In practice, it does not matter very much whether it's *loess* or *lowess*, or the polynomial order (p).

The span matters:

- ✓ very low ⇒ low bias, but high variance: "undersmoothing";
- ✓ very high ⇒ high bias, but low variance: "oversmoothing".

A middle ground has to be found by the researcher, but with erring on the side of "undersmoothing" (Keele, 2008, p. 34).[2]

[2] Same with polynomial order: better fit vs. extra parameters.

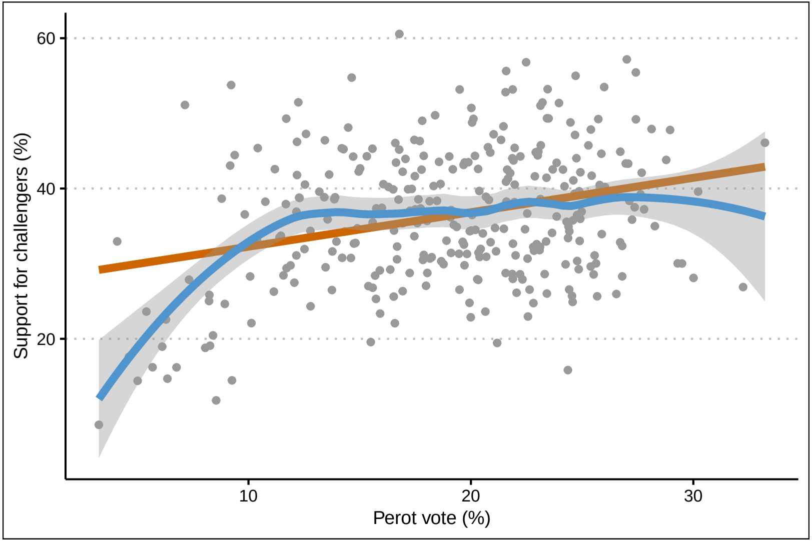
Inference

Since at each step a regression is run, inference based on SEs can be easily conducted.[3]

This is usually displayed on the plot, in the form of confidence intervals around the line.

[3] The formulas for this are no longer nice, so I omit them here, but you can get a quick look at them in Keele (2008, pp. 39–41)

Inference for Perot vote



LPR with span of 0.50 and SEs

Hypothesis testing

An even more powerful application is to test whether the nonparametric model fits the data better than the parametric linear one.

We can do this as the latter model is a restricted version of the former.

$$F = \frac{\frac{RSS_0 - RSS_1}{J}}{\frac{RSS_1}{df_{res}}} \tag{6}$$

$df_{res} = n - p_1$, where p_1 is the effective number of parameters of the smoother, and n is the sample size.

df_{res} need not be an integer in the case of smoothers.

Hypothesis testing

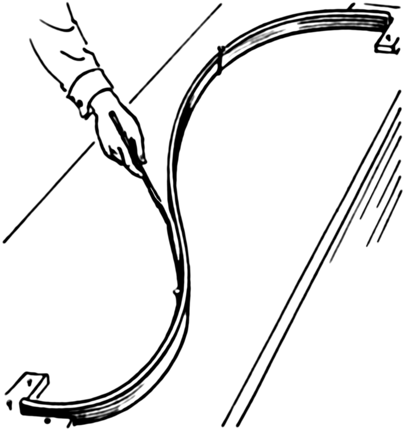
The other terms in the formula:

- ✓ RSS_0: residual sum of squares from restricted specification;
- ✓ RSS_1: residual sum of squares from nonparametric specification;
- ✓ $J = p_1 - p_0$: difference in effective number of parameters

In this case, the F-test $= 4.742065$, and $p < 0.001$, suggesting that the nonparametric model fits the data better.

Regression Splines

Advantages of splines



- ✓ will provide the best mean squared error fit;
- ✓ a smoothing spline is designed to prevent overfitting;
- ✓ easier to incorporate in semiparametric models than LPR.

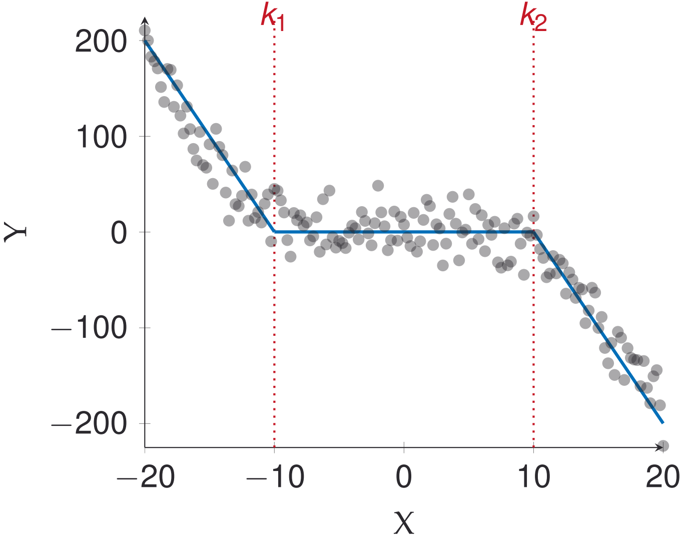
The use of *knots*

Splines are essentially polynomials fitted to separate regions of the data.

The “borders” between these regions are called *knots* (just values of X).

The polynomials are fit to the separate regions, and forced to meet at the *knot*.

Piece-wise polynomials



Very simple example, with 1st order polynomial and 2 knots

Piece-wise polynomials

The number of knots, their position, as well as the degree of the polynomial specification are chosen by the researcher.

For most realistic problems, only 4–5 knots are really needed.

Let's take the situation of a single knot:

$$Y_i = a + b_1 X_i + b_2 (X_i)_+ + e_i \tag{7}$$

Piece-wise polynomials

This $(X)_+$ is a new variable, which we obtain from X, based on its position with respect to c_1 (the knot position).

$$(x_i)_+ = \begin{cases} x_i, & \text{for } x_i > c_1 \\ 0, & \text{for } x_i \leq c_1 . \end{cases} \tag{8}$$

$$Y_i = \begin{cases} a + b_1 X_i, & \text{for } x_i \leq c_1 \\ a + b_1 X_i + b_2 (X_i - c_1), & \text{for } x_i > c_1 . \end{cases} \tag{9}$$

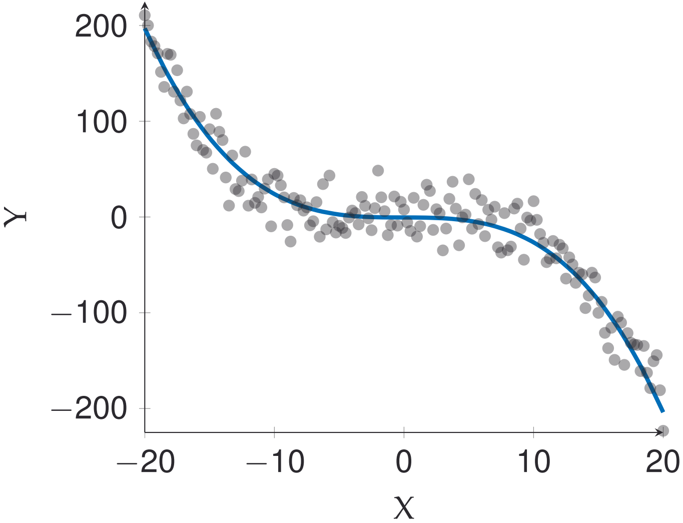
More complex forms

So far, I have only used linear specifications, as they make the formulas accessible.

However, we can easily specify quadratic and cubic specifications, in case the data patterns reveal such forms are needed.

- ✓ knots can be added by default at the lowest and highest data points, so as to fit cubic splines in these regions as well: *natural splines*;
- ✓ piece-wise functions can be rescaled, so as to avoid collinearity between X and $(X)_+$: *B-splines*.

More complex forms



Knot placement and

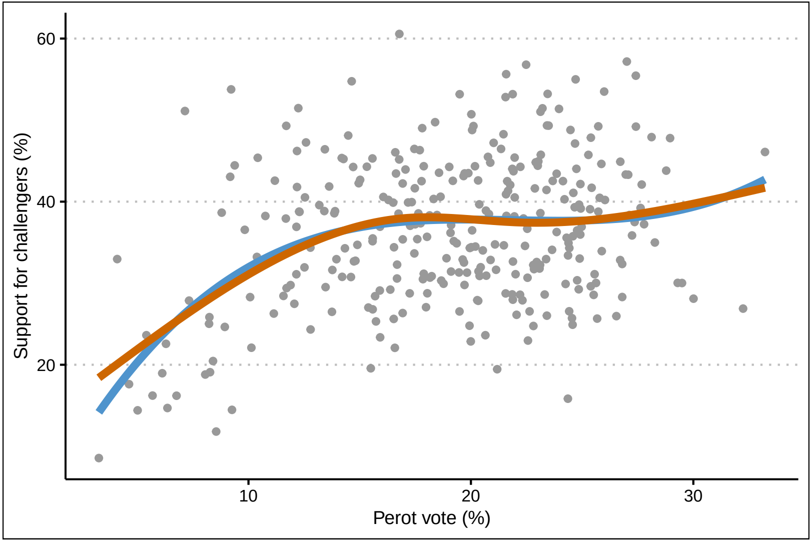
Usually placed at equal intervals in the data, e.g. quartiles or quintiles.

The number of knots is the more important choice—it governs how smooth the final fit will be.

2 methods:

- ✓ visual: start with 4 knots, and increase/decrease number if the fit is too smooth/rough;
- ✓ statistical: use the AIC of the fit, and select the number of knots that produces the lowest AIC.

Splines for Perot vote



Cubic B-spline and natural spline fits

Smoothing splines

Penalized splines

One frequent accusation is that it's very easy to overfit the data with splines, as you can simply select a large number of knots.

Penalized (smoothing) splines are a solution to this, as they introduce a penalty for every additional parameter estimated.[4]

$$SSR = \sum_{i=1}^{n} [Y - f(X)]^2 \tag{10}$$

In standard linear regression this $f(X)$ is the model specification.

[4] In the same way that the adjusted R^2 includes such a penalty.

Penalized splines

For penalized splines, a modified version of SSR gets minimized.

$$SSR^{*} = \sum_{i=1}^{n}[Y - f(X)]^{2} + \lambda \int_{x_1}^{x_n}[f^{''}(x)]^{2}dx \qquad (11)$$

$f^{''}(x)$ is the second derivative to the nonlinear fit. The less smooth the curve, the higher this second derivative is.

λ is called the smoothing (tuning) parameter. The higher it is, the smoother the fit (but, also, more biased).

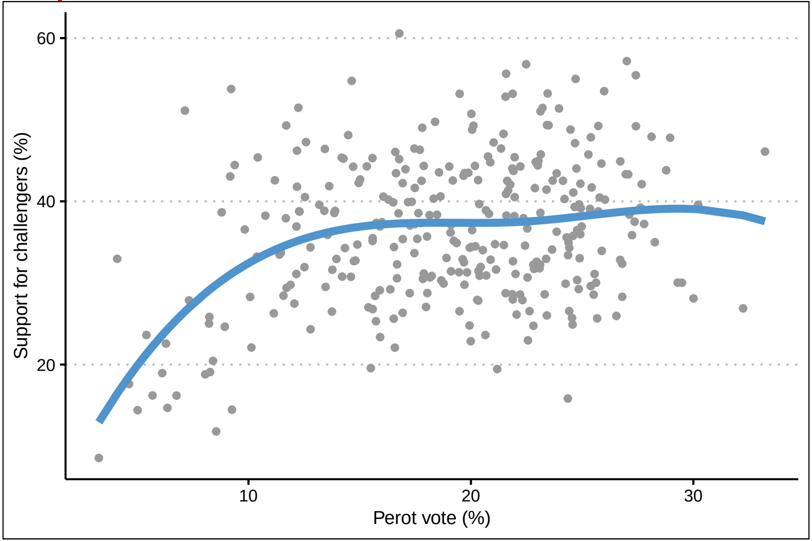
Penalized splines

Because of the smoothing parameter, λ, the number of *knots* does not matter that much anymore.

Allowing for any order of derivative, $f^m(x)$, produces what are called *thin plate splines*. These are useful for smoothing in larger number of dimensions.

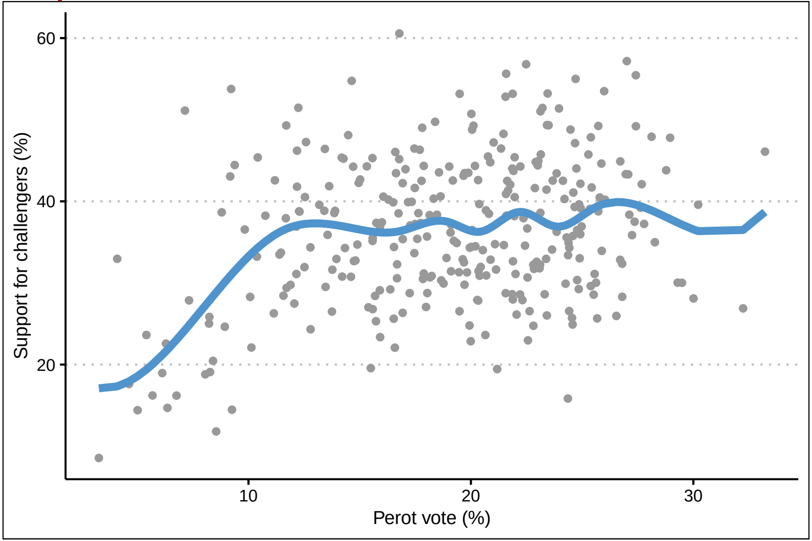
Constructing confidence intervals and hypothesis tests (with the F-test) proceeds the same as with standard splines based on piece-wise polynomials.

Penalized splines



Smoothing spline with 4 knots and $\lambda = 0.00179$.

Penalized splines



Smoothing spline with 10 knots and $\lambda = 0.00179$.

Thank you for the kind attention!

References I

Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, *74*(368), 829–836.

Keele, L. (2008). *Semiparametric Regression for the Social Sciences*. Chichester, UK: Wiley.