Introduction to Data Science for Public Policy

Class 10: Non-linear models

Thomas Monk

t.d.monk@lse.ac.uk

adapted from Introduction to Statistical Learning, (http://www.rnfc.org/courses/isl/), and with thanks to Dr. Jack Blumenau, UCL.

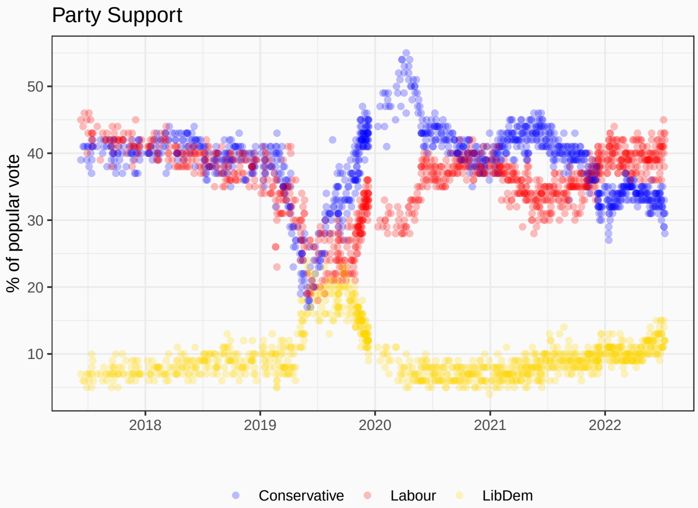
September 9 2022

Motivation

How popular are the UK political parties?

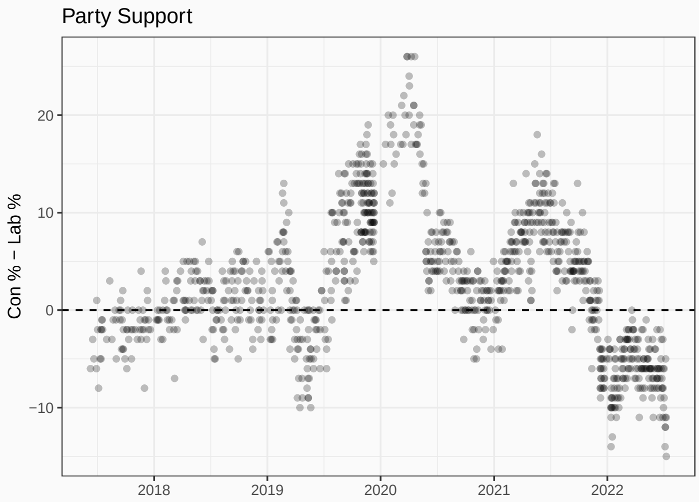
The last few years have seen dramatic changes in UK politics, most of which has been tracked closely by a variety of political polling companies. Individual polls are noisy manifestations of underlying trends in public opinion. We can think of the task of measuring public support for a given party as a prediction problem, where results in each polls are data and we want to predict the true average level of support at each point in time. What is the best model for predicting such trends?

Motivation



Party Support
% of popular vote
50
40
30
20
10
2018
2019
2020
2021
2022
● Conservative
● Labour
● LibDem

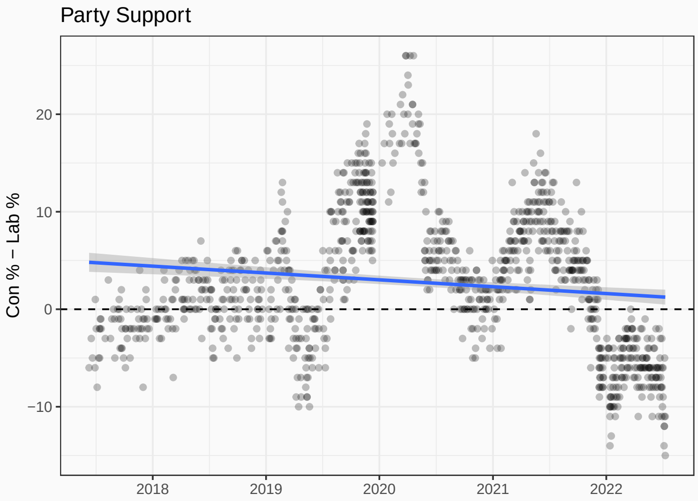
Motivation



Motivation

- We are interested in describing how the Conservative lead varies over time.
- What is an appropriate model for this data?
- We could, of course, use a linear regression.

Motivation



This is a dumb model for this data.

Motivation

- Traditional models – like linear and logistic regression – can be good for evaluating theories that imply specific functional forms for the relationship between outcomes and predictors.

Motivation

- Traditional models – like linear and logistic regression – can be good for evaluating theories that imply specific functional forms for the relationship between outcomes and predictors.
- For many social science domains, the theories we have are not sufficiently detailed to account for the complexity of the data we have to hand.

Motivation

- Traditional models – like linear and logistic regression – can be good for evaluating theories that imply specific functional forms for the relationship between outcomes and predictors.
- For many social science domains, the theories we have are not sufficiently detailed to account for the complexity of the data we have to hand.
- We therefore need a set of methods which allow us to learn complicated non-linearities and interactions from the data.

Basis functions - can skip

- Many of the approaches we discuss today are special cases of what are known as “basis function” approaches.
- They share the idea that, in order to capture non-linear relationships between predictors and outcome, we can transform X in some way and then just use a linear model.
- The interpretation of these is exactly as we had in PP445.

Polynomial regression

Polynomial regression models

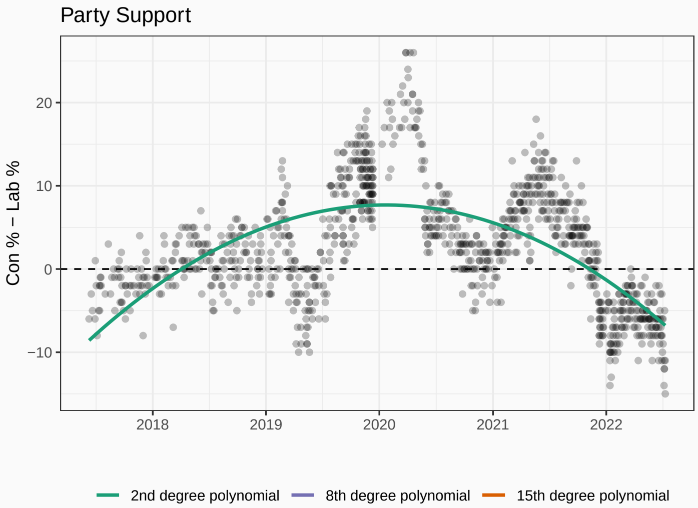
Polynomial regression models take the following form:

$$\begin{aligned}
\text{Linear:} \quad y_i &= \alpha + \beta_1 x_{i1} + \epsilon_i \\
\text{Quadratic:} \quad y_i &= \alpha + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \epsilon_i \\
\text{Cubic:} \quad y_i &= \alpha + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \beta_3 x_{i1}^3 + \epsilon_i \\
\text{General:} \quad y_i &= \alpha + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \ldots + \beta_d x_{i1}^d + \epsilon_i
\end{aligned}$$

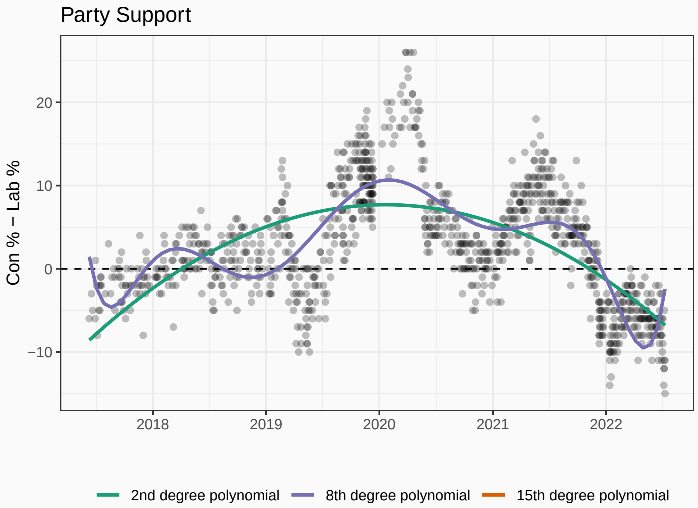
Where x_{i1}^2 is just $x_{i1} \cdot x_{i1}$ and x_{i1}^3 is just $x_{i1} \cdot x_{i1} \cdot x_{i1}$, and so on (i.e. the basis function for the quadratic is $b(x_i) = x_i^2$).

The more polynomial terms we add, the more flexible we are allowing the relationship between our outcome and our predictor to be.

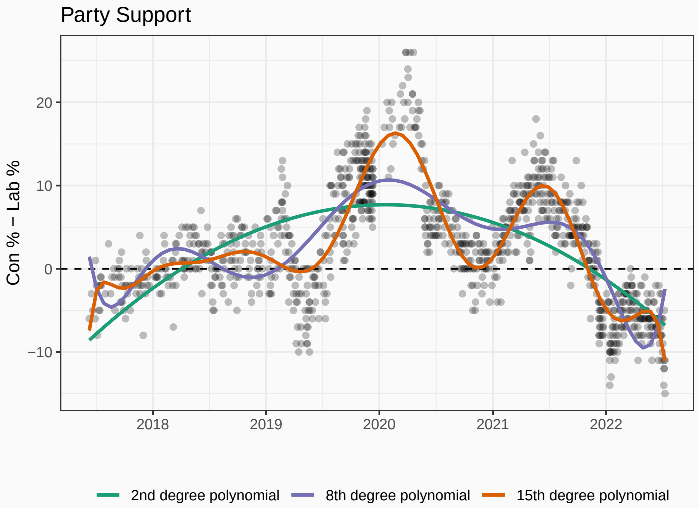
Polynomial regression – application



Polynomial regression – application



Polynomial regression – application



Polynomial regression

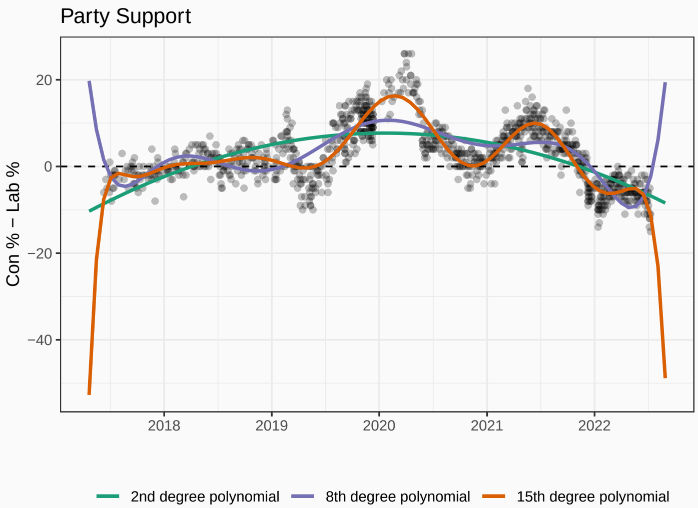
Advantages:

- Very easy to implement: e.g. `poly(x, degree = 3)`
- Can be used in any regression framework (for instance, logit)

Disadvantages:

- Can have very poor behaviour in the tails, which makes it a poor tool for extrapolation beyond the domain of X
- For example, let's extend the x-asis of our party support date by 50 days in either direction...

Polynomial regression – extrapolation



Step Functions

Step function regression models

Step function, or piecewise-constant, models take the following form:

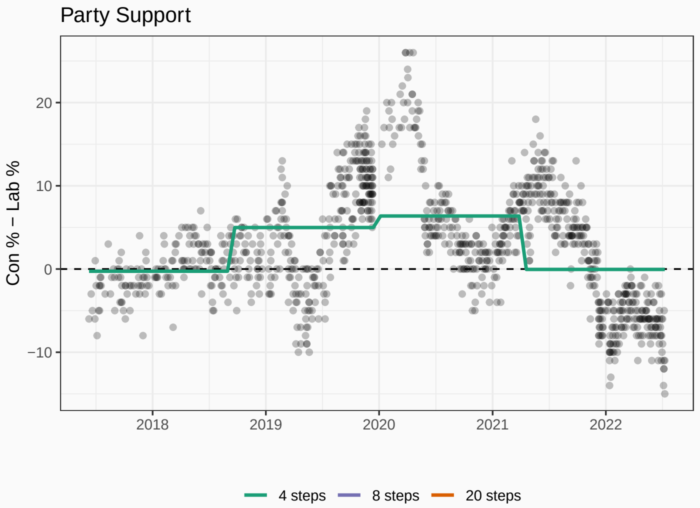
$$y_i = \alpha + \beta_1 C_1(x_{i1}) + \beta_2 C_2(x_{i1}) + \ldots + \beta_d C_d(x_{i1}) \epsilon_i$$

Where we transform x into a set of dummy variables by defining a set of cutpoints, $c_1, c_2, \ldots, c_k$ across the range of x:

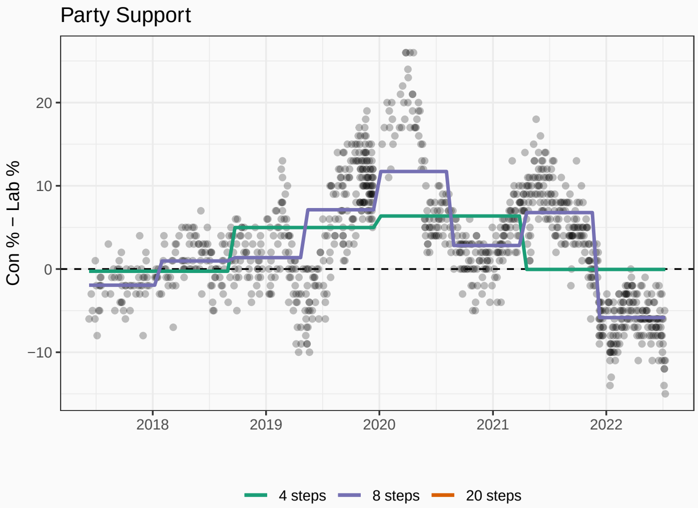
$$\begin{aligned}
C_0(x) &= I(X < c_1) \\
C_1(x) &= I(c_1 \leq x < c_2) \\
C_2(x) &= I(c_2 \leq x < c_3) \\
&\ldots \\
C_K(x) &= I(c_k \leq x)
\end{aligned}$$

The more cutpoints we add, the more flexible we are allowing the relationship between our outcome and our predictor to be.

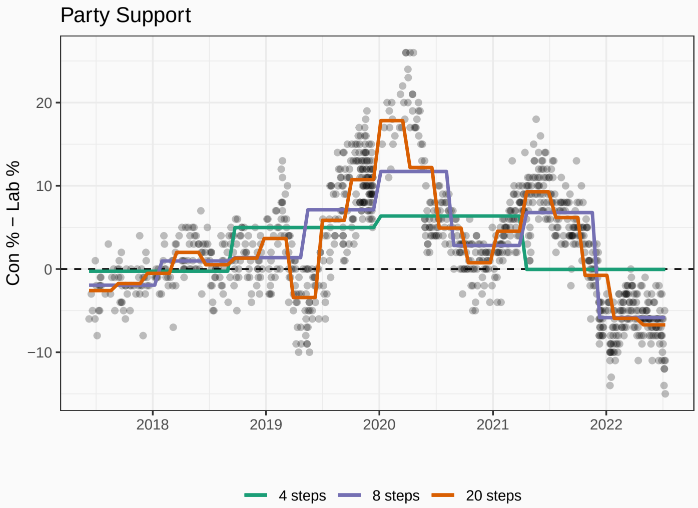
Step functions – application



Step functions – application



Step functions – application



Splines – Piecewise polynomials

- Polynomial models: specify a single polynomial across the domain of X
- Step function models: estimate a single mean parameter for different ranges of X
- Splines: combine both approaches by specifying separate polynomial models for different regions of X

Splines – Piecewise polynomials

Piecewise polynomial models

Regression splines generalise the step-function approach above by allowing for more flexible functional forms for different ranges of X:

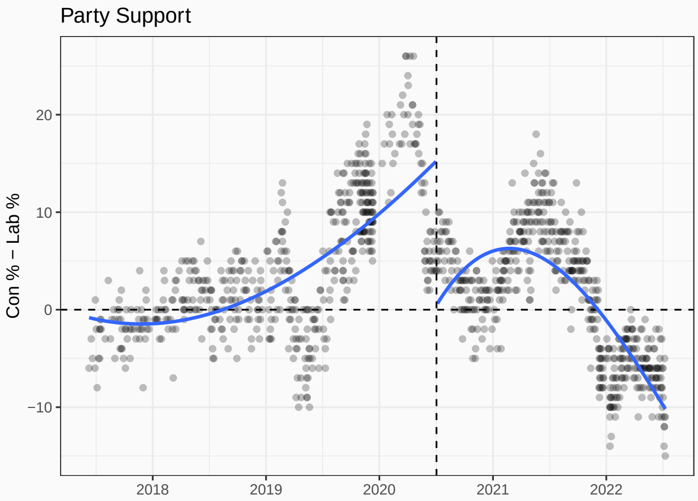
$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

where c is a ”knot” which defines a point in X at which the coefficients change.

More cutpoints, and higher order polynomials, imply a more flexible relationship between our outcome and our predictor. E.g.

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c_1; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } c_1 \geq x_i < c_2 \\ \beta_{03} + \beta_{13}x_i + \beta_{23}x_i^2 + \beta_{33}x_i^3 + \epsilon_i & \text{if } x_i \geq c_2. \end{cases}$$

Splines – Piecewise polynomials

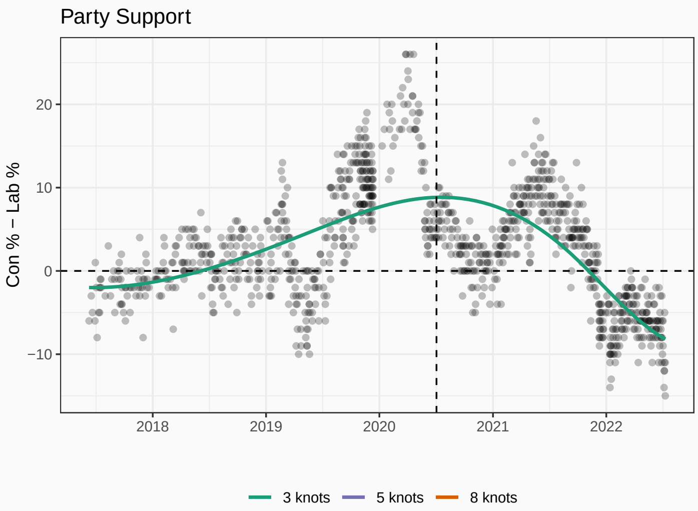


What is wrong here?

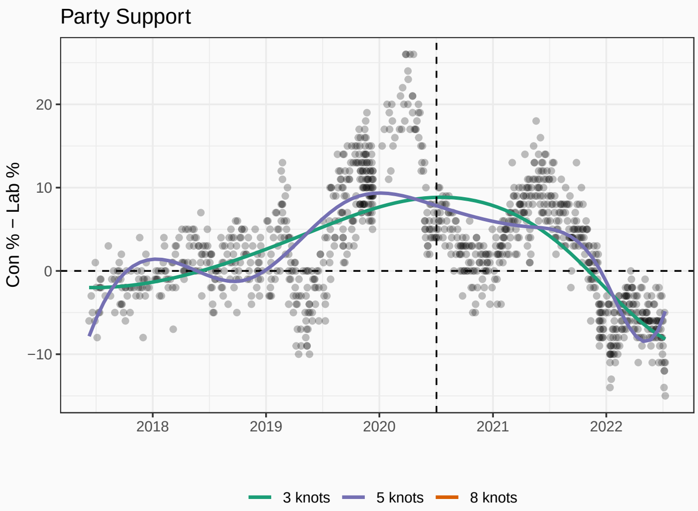
Splines – Piecewise polynomials

- Even though we would like to model separate polynomials on either side of the knot/cutpoint, we don't want a big discontinuity at that point
- Instead, we would like to constrain the estimates so that
    - There is no *disconuity* at the knots
    - The piecewise polynomials are *smooth* at the knots
- We can constrain the estimation to achieve these properties by using truncated basis functions of X

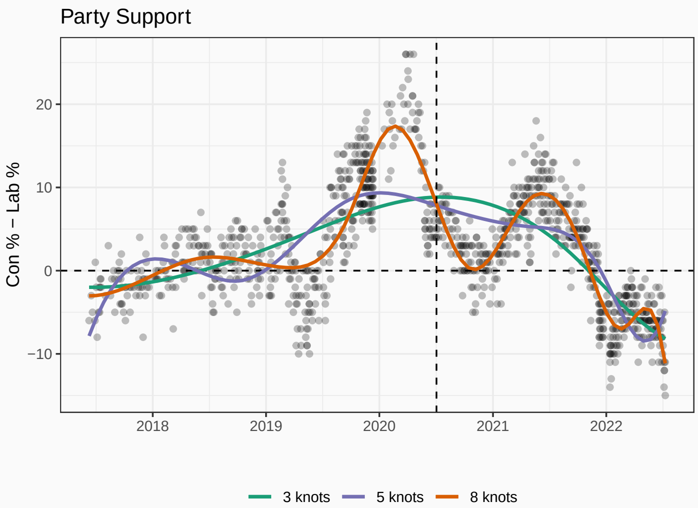
Splines – Piecewise polynomials



Splines – Piecewise polynomials



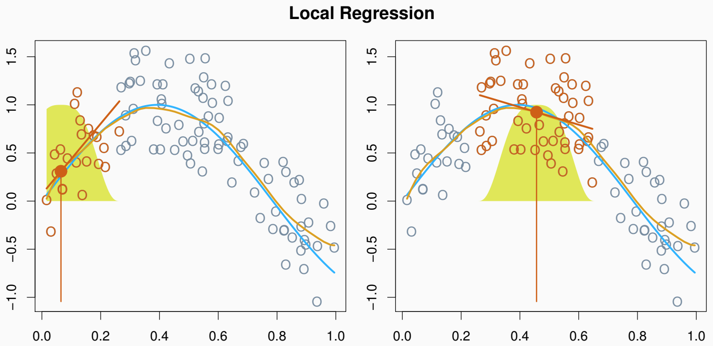
Splines – Piecewise polynomials



Local linear regression

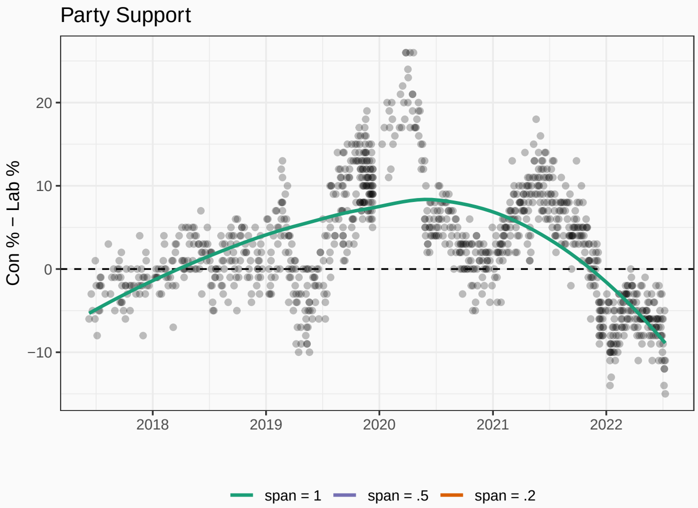
- Local regression models do not account for non-linearities not by transforming x using a basis function
- Instead, they estimate the relationship between x and y at different points in the range of x
- For each point, x_0, we fit a weighted linear regression model using only those observations that are within some distance of x_0
- Observations that are closer to x_0 are assigned a higher weight in determining the local regression slope than observations further away
- The key parameter here is the *span*, s, which controls the proportion of points used to estimate the local regression
    - High values of $s \longrightarrow$ less flexible model
    - Low values of $s \longrightarrow$ more flexible model

Local linear regression

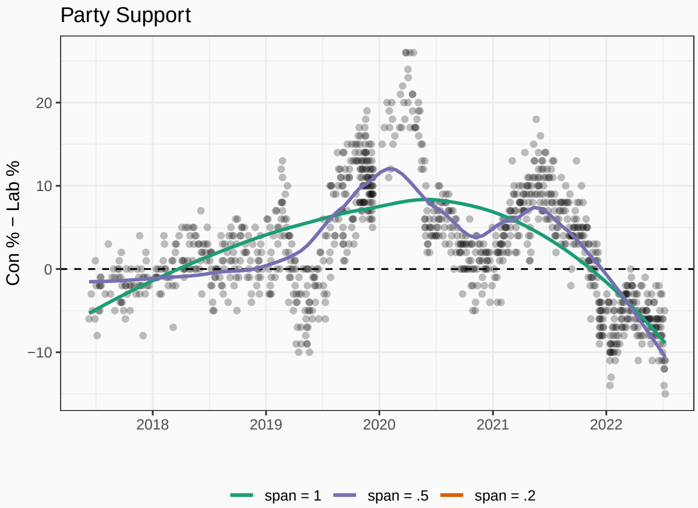


- With a sliding weight function, we fit separate linear fits over the range of X by weighted least squares.
- As s decreases, so does the size of the sliding window

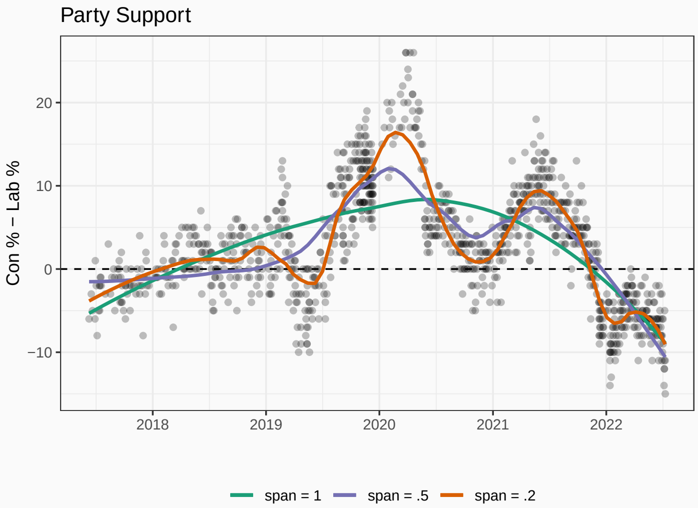
Local linear regression – application



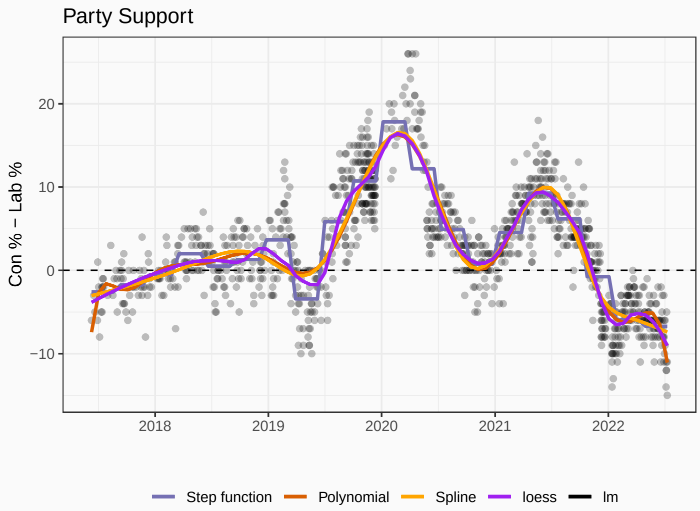
Local linear regression – application



Local linear regression – application



Which Wiggle is Best?

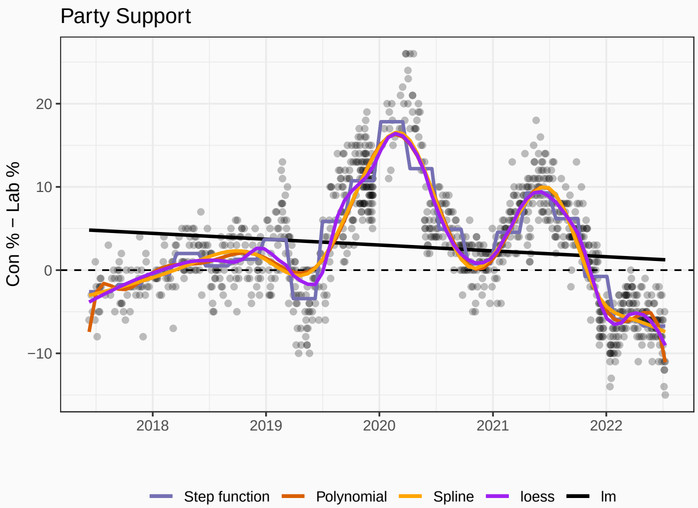


Choices, choices

Note that each of these methods requires the researcher to make modelling decisions which are consequential for prediction:

1. **Polynomial regression**
   - Choices: Order of polynomial (X^2, X^2, X^3, etc)
2. **Step functions**
   - Choices: Number of steps; where they are placed
3. **Splines**
   - Choices: Number of knots; where they are placed; order of polynomial
4. **Local regression**
   - Choices: Span
5. **Generalised additive models**
   - Choices: Possibly all of the above!

Easier Question: Which Wiggle is Worst?



Generalized Additive Models

One of the great things about linear regression (or, regression in general) is that we can include *many* predictors:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i.$$

Generalised additive models provide a framework in which we can incorporate flexible non-linearities for several variables into the additive structure of linear models:

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

“Additive” because we calculate f_j for each X_j and then add them together!

Decision Trees

Frame Title

Predicting women’s wages

Can we predict how much women earn? Using data from the National Longitudinal Survey of Youth, we will try to predict women’s wages from a set of individual and occupational characteristics. In addition to building our predictive model, we will focus on which variables are most important for constructing these predictions.

Tree-based Methods

While we could use linear regression to predict this outcome, here we will focus on tree-based methods.

Approach:

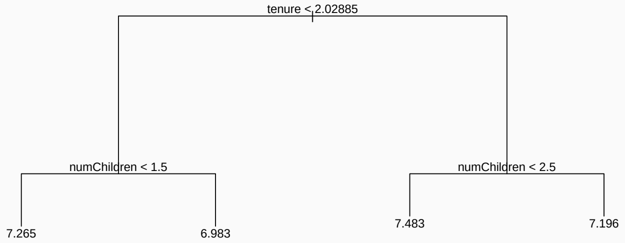
1. Stratify or segment the predictor space into a number of simple regions
2. Calculate the mean outcome in each region
3. Predict $\hat{y}_i$ on the basis of the region into which i falls

Since the set of splitting rules used to segment the predictor space can be summarised in a tree, these types of approaches are known as decision-tree methods.

Pros and Cons

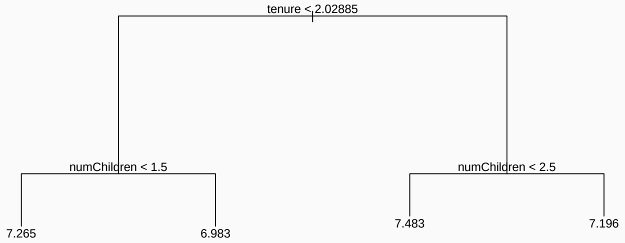
- Tree-based methods are simple and useful for interpretation.
- However they typically are not competitive with the best supervised learning approaches in terms of prediction accuracy.
- Hence we also discuss bagging, and random forests. These methods grow multiple trees which are then combined to yield a single consensus prediction.
- Combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss interpretation.

A Simple Tree



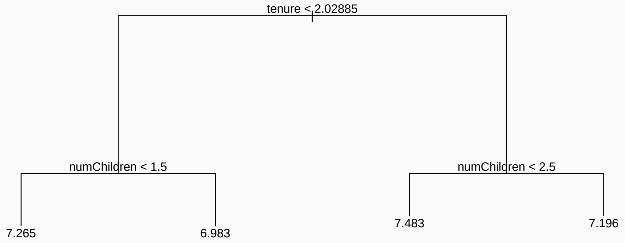
A regression tree predicting log wage as a function of tenure and number of children

A Simple Tree



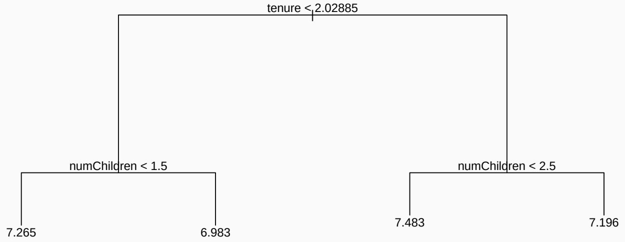
At any internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$

A Simple Tree



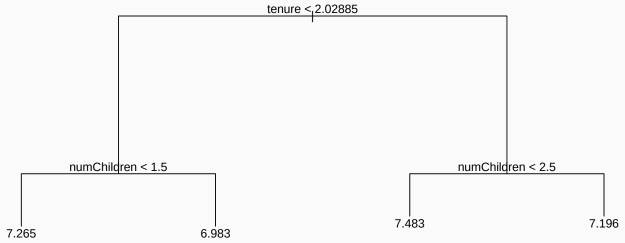
Here, the left-hand branch corresponds to $tenure < 2.02$, and the right-hand branch corresponds to $tenure \geq 2.02$

A Simple Tree



The points along the tree where the predictor space is split are referred to as internal nodes.

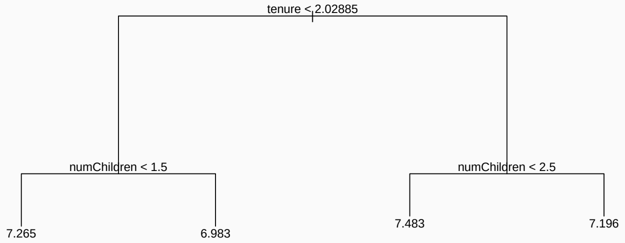
A Simple Tree



In this tree, three internal nodes are indicated:

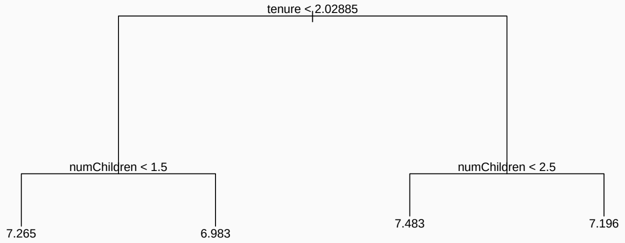
- $tenure < 2.02$
- $numChildren < 1.5$
- $numChildren < 2.5$

A Simple Tree



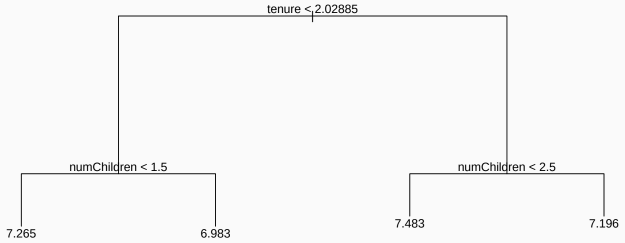
Note that the splitting rule is different for the two internal nodes involving `numChildren`! (More on this later)

A Simple Tree



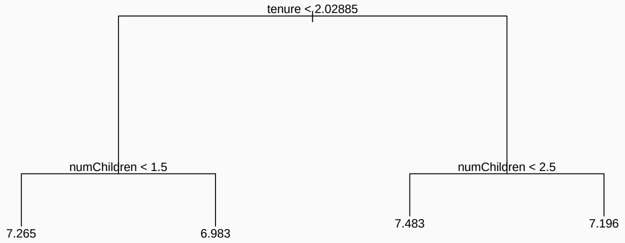
The terminal nodes, or *leaves*, indicate the mean of Y for the observations in that partition.

A Simple Tree



We use the values in the leaves as the predicted values, $\hat{y}$, for new observations.

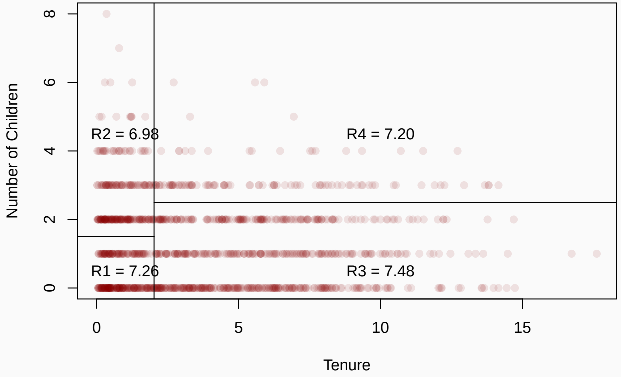
A Simple Tree



Interpretation:

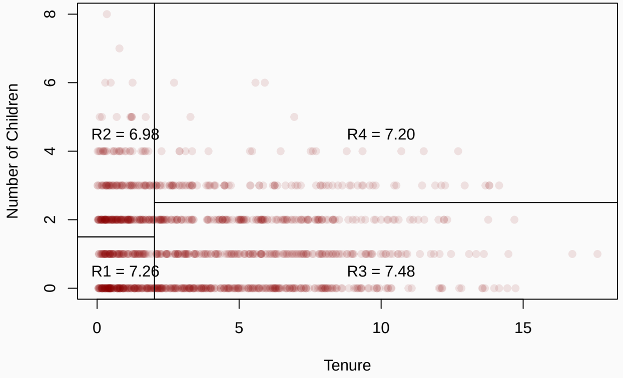
- $tenure$ is the most important factor in determining $wage$, and individuals with less experience earn lower salaries than more experienced individuals
- Conditional on $tenure$, people with fewer children earn more than people with more children

Results



The tree stratifies or segments individuals into four regions of predictor space

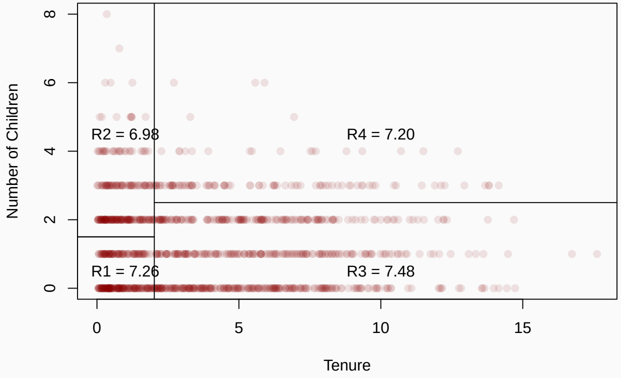
Results



$$R_1 = \{X | Tenure < 2.02, Children < 1.5\}$$

$$R_2 = \{X | Tenure < 2.02, Children > 1.5\}$$

Results



$$R_3 = \{X | Tenure > 2.02, Children < 2.5\}$$

$$R_4 = \{X | Tenure > 2.02, Children > 2.5\}$$