# Overview of Nonparametric Regression

Chad M. Schafer

Department of Statistics & Data Science, Carnegie Mellon University

cschafer@cmu.edu

April 2021

# Outline

Motivation

Underlying Concepts

Nonparametric Approaches

Some Theory

The Curse of Dimensionality

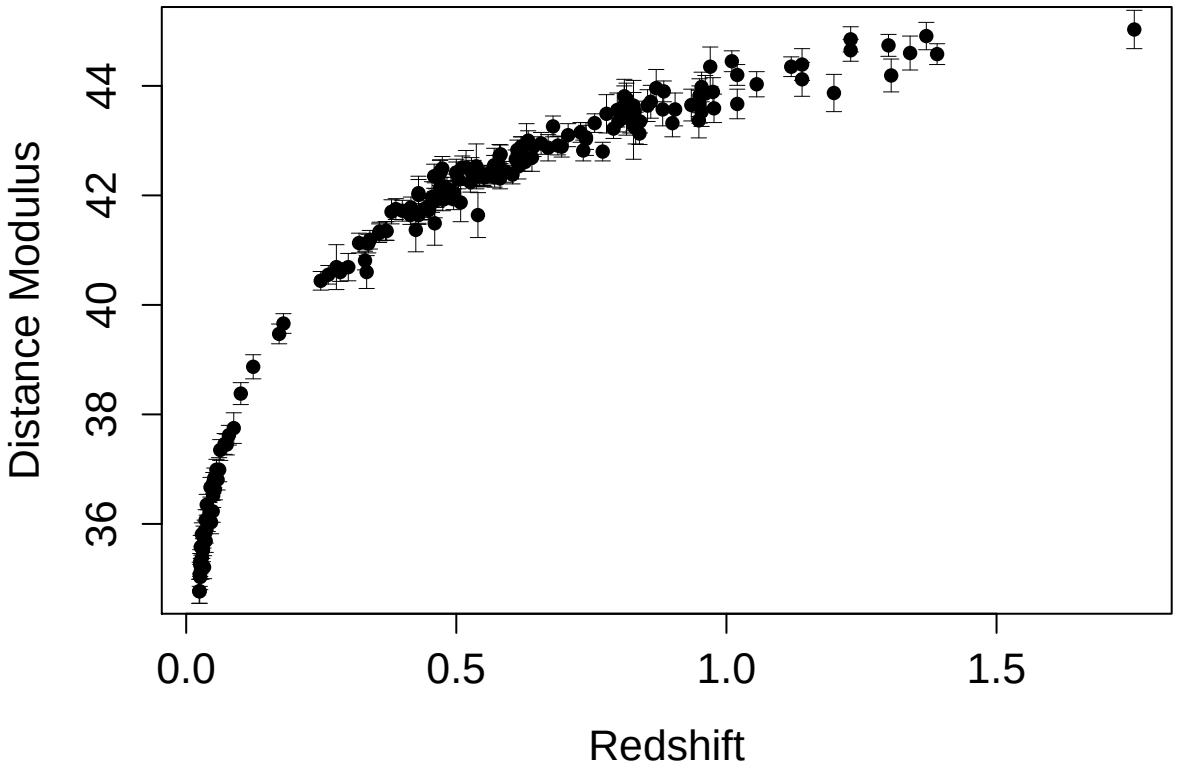
Additive Models

# Motivation

The objective: Construct a **model** that relates the **response variable** $Y$ to the **predictor variables** $X_1, X_2, \ldots, X_p$.

**Example:** Modelling relationship between distance modulus and redshift for Type Ia Supernovae

$$\begin{aligned} Y &= \text{distance modulus} \\ X_1 &= \text{redshift} \end{aligned}$$

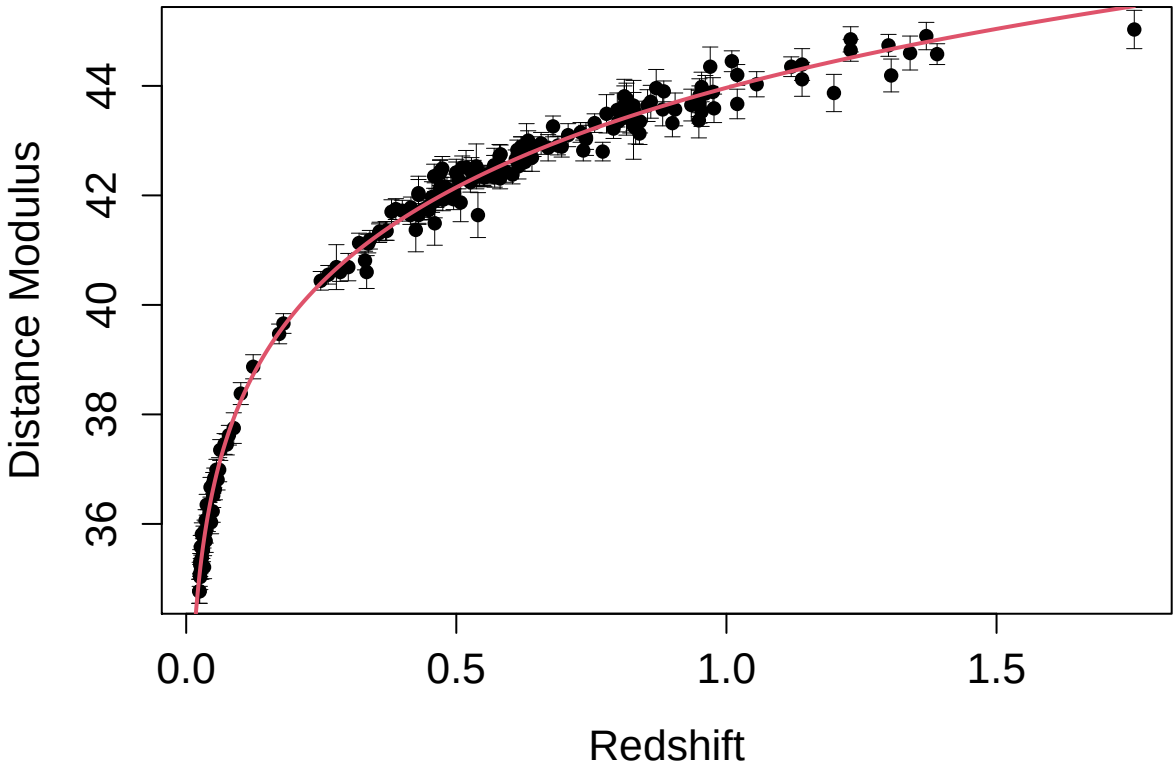



From Riess, et al. (2007), 182 Type Ia Supernovae

The $\Lambda$CDM model proposes a simple **parametric** model for the relationship between redshift $(z)$ and distance modulus:

$$\text{Distance Modulus} = 5\log_{10}\left(\frac{c(1+z)}{H_0}\int_0^z \left(\Omega_m(1+u)^3 + (1-\Omega_m)\right)^{-1/2}\right) + 25$$

This depends on two **cosmological parameters**, $\Omega_m$ and $H_0$. Hence learning this relationship can help constrain these parameters.



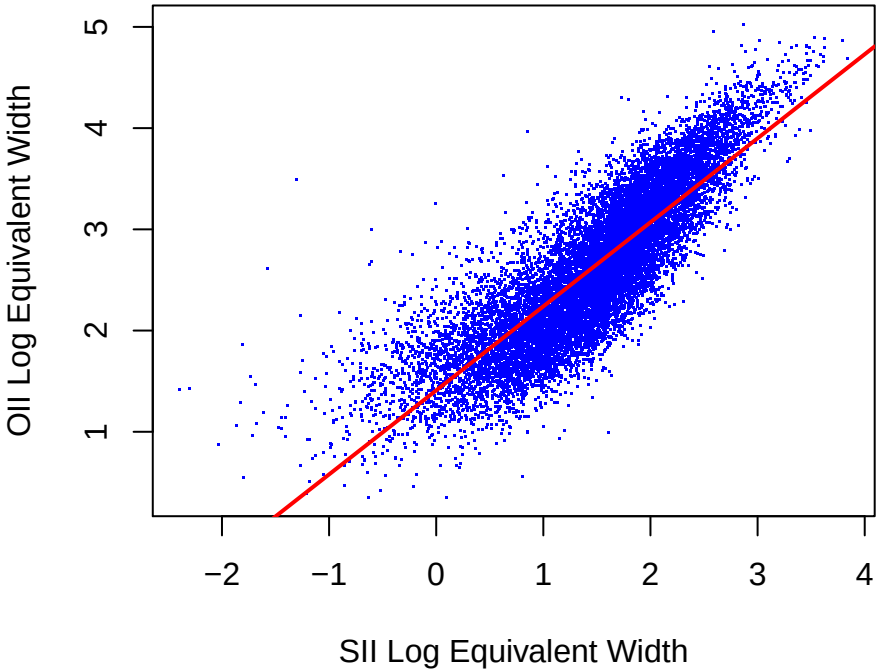

The case where $H_0 = 72.76$ and $\Omega_m = 0.34$.

**Example: Modelling Emission Lines**

Here we seek to model the relationships between the log equivalent widths of different emission lines of galaxies. This was motivated by an effort to improve simulation models for galaxy spectra.

The sample consists of 13,788 SDSS galaxies.

A scatter plot showing the relationship between a pair of line strengths:




In this example, the objective is **not** parameter estimation.

# Underlying Concepts

**Linear regression** models predict the response variable as a **linear combination** of the predictors, plus random scatter

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

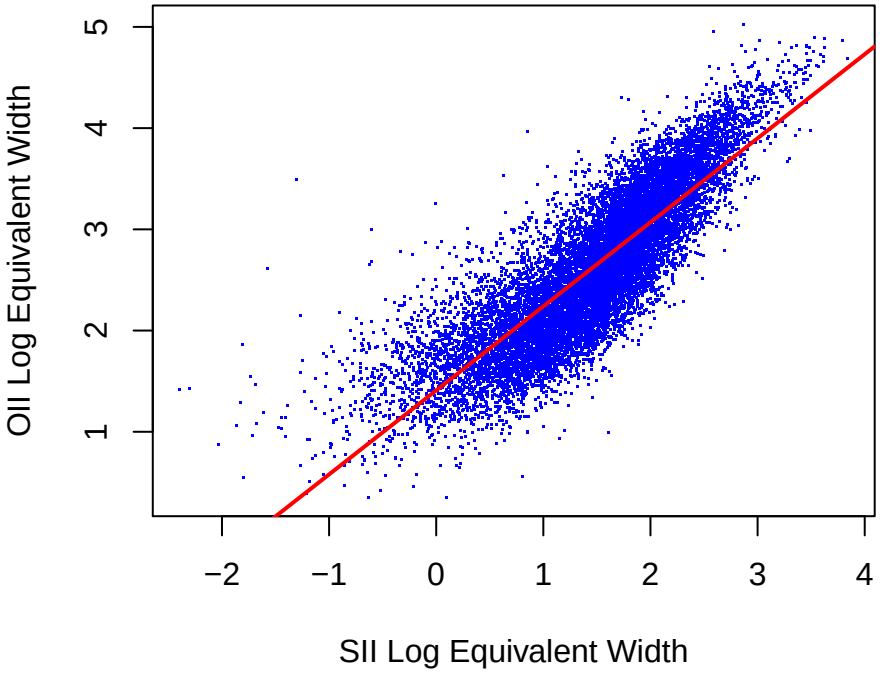
These can be more flexible than initial impressions may suggest.

For example, this describes a cubic relationship between $X$ and $Y$:

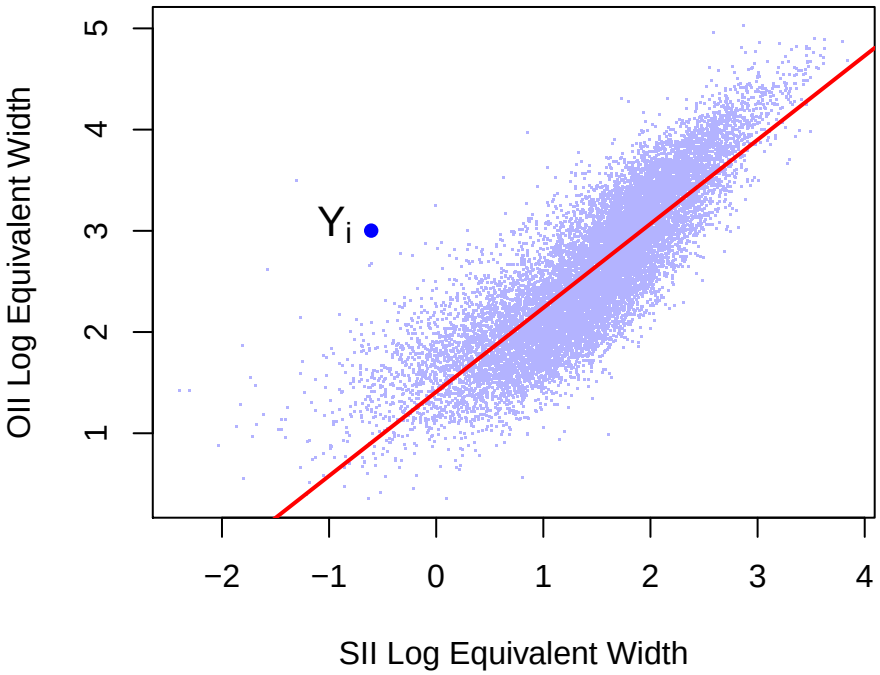
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon,$$

but it is a "linear model" since it is linear in the $\beta$ parameters.

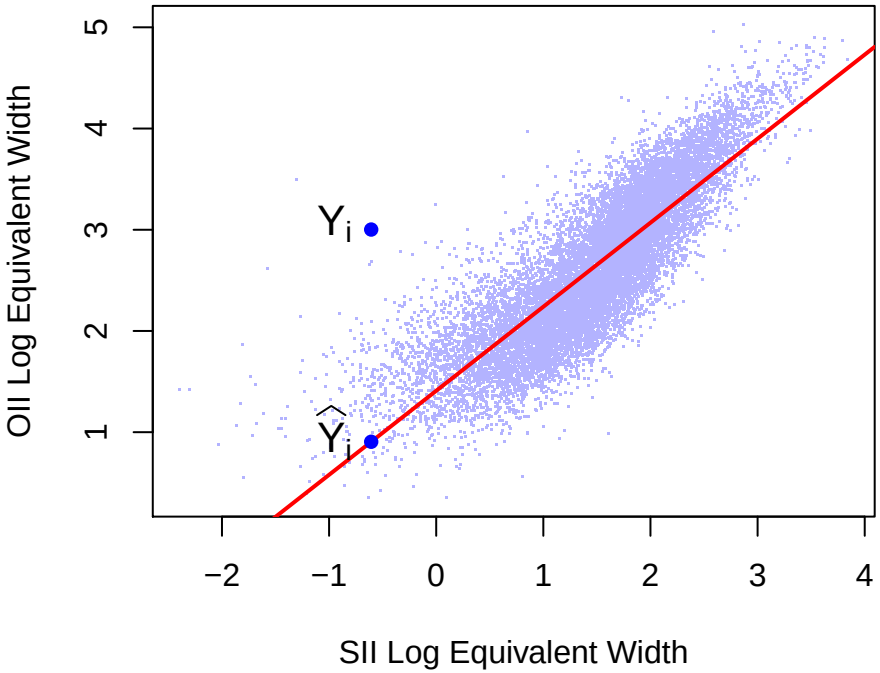
A **simple linear regression** fit using **least squares**:



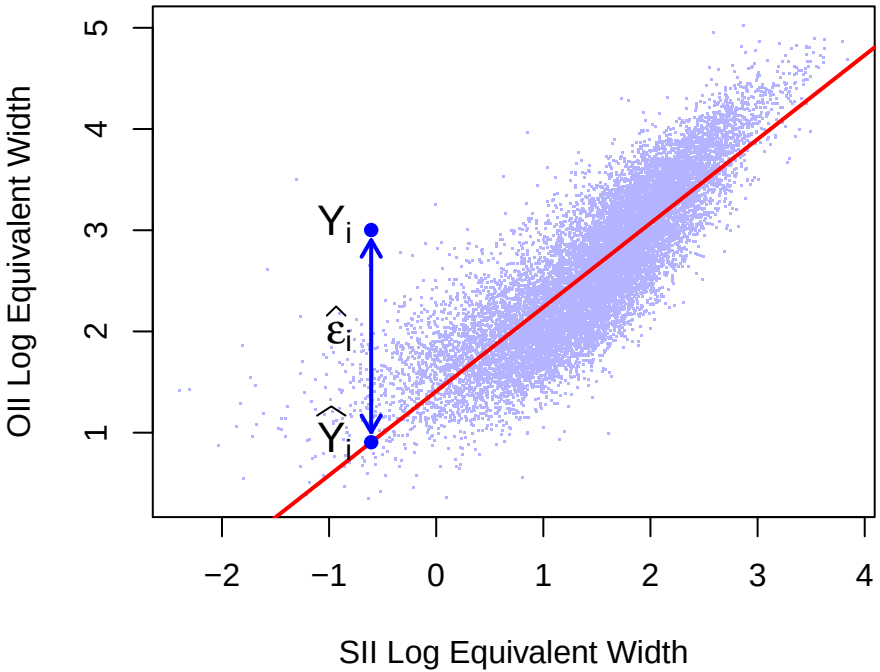
Consider one observation in the data set, $Y_i$:



The prediction from the regression model is the **fitted value** $\widehat{Y}_i$:



The difference is the **residual**, $\widehat{\epsilon}_i = Y_i - \widehat{Y}_i$:




The least squares regression line minimizes $\sum_i \widehat{\epsilon}_i^2$.

**Residuals as Diagnostic Tools**

We cannot rely upon purely numerical assessments to judge the quality of a model fit.

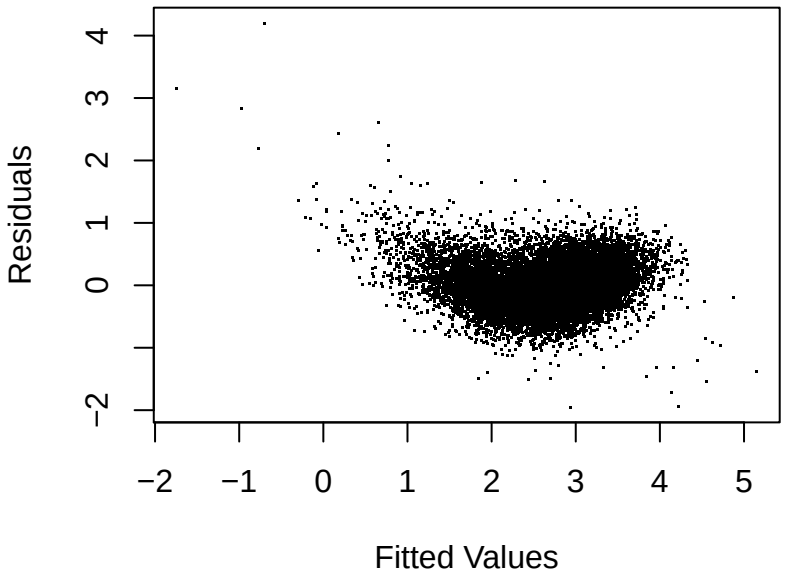
Inspection of the residuals is an important diagnostic tool.

The $i^{th}$ residual is an approximation to the $i^{th}$ **irreducible error** $(\epsilon_i)$:

$$\begin{aligned}
\widehat{\epsilon}_i &= Y_i - \widehat{Y}_i \\
&= Y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 X_{1i} + \cdots + \widehat{\beta}_p X_{pi}) \\
&\approx Y_i - (\beta_0 + \beta_1 X_{1i} + \cdots + \beta_p X_{pi}) \\
&= \epsilon_i
\end{aligned}$$

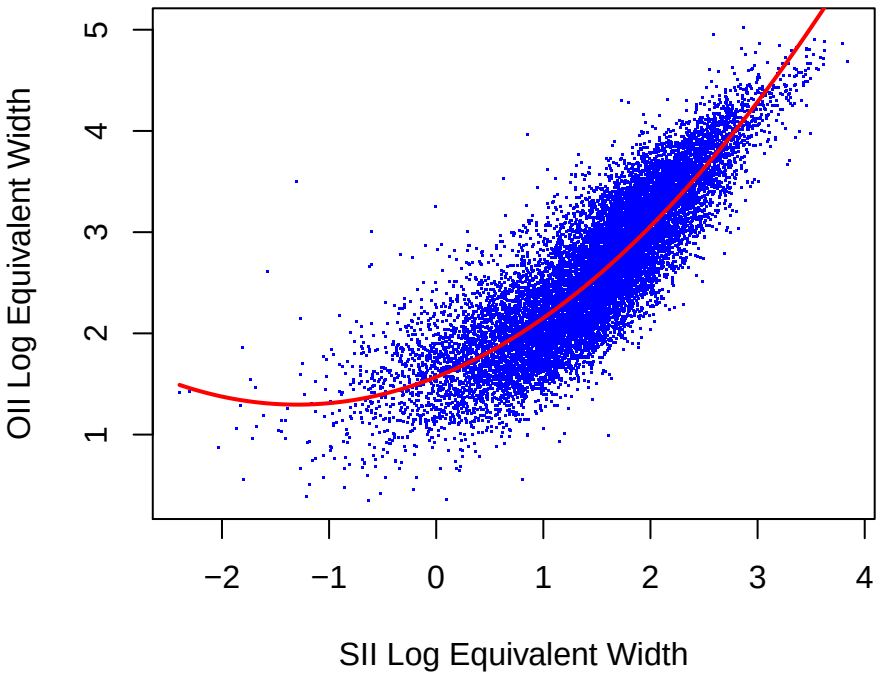
**If** the model is a good fit, then the residuals should be random scatter, and lacking in any relationships with predictors, etc.

The plot of **residuals versus fitted values** is standard. The curved shape to the plot below shows a lack of fit in our linear regression example.



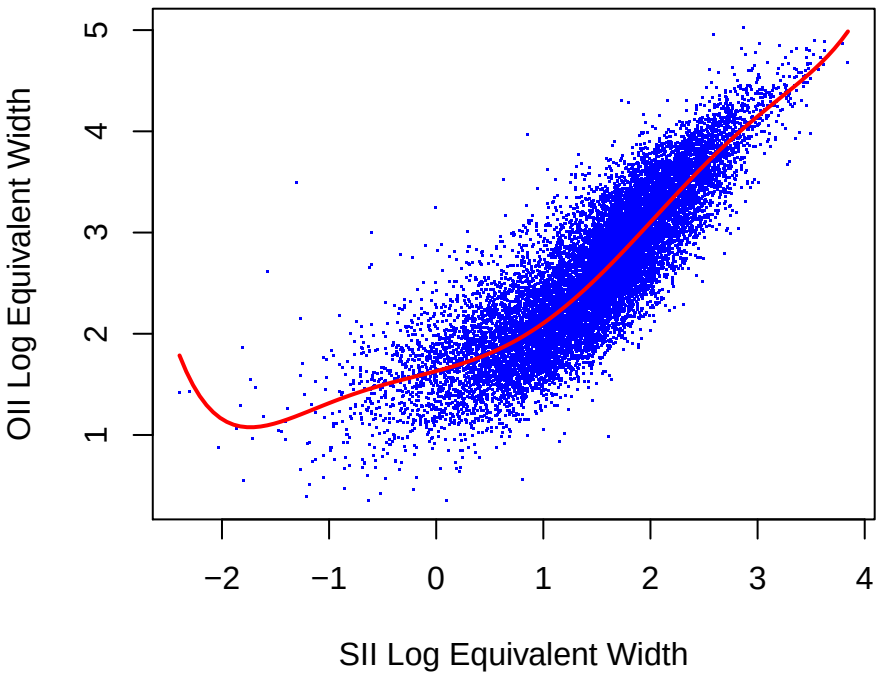
A quadratic fit:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$



Including up to sixth power:

$$Y = \beta_0 + \sum_{k=1}^{6} \beta_k X^k + \epsilon$$



## Model Selection

The process of **model selection** involves deciding which of the available predictors should be utilized in the model.

Including too many predictors leads to **overfitting**, a situation in which a model fits better to the training sample than it would to external observations not used in the fitting. This is a big problem.

**Important:** Increasing the number of predictors will necessarily decrease the sum of the squared residuals. [1] This is **not** useful for model selection.

[1] Ignoring some technical cases where the new predictor is a linear combination of the predictors already included.

**Leave-one-out cross-validation** is an important tool for model selection.

For each observation $i$, imagine fitting a model which "leaves out" that row of the data set. The model is fit on this reduced training set.

The fit model is then used to predict the response for observation $i$. Denote this $\widehat{Y}_{(-i)}$. The difference between $Y_i$ and $\widehat{Y}_{(-i)}$ is a useful view of the performance of the model at predicting the response value for "new" cases.

We can accumulate these errors into a score:

$$\text{LOOCV Score} = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \widehat{Y}_{(-i)} \right)^2$$

This score can be compared over models with different sets of predictors. Of course, models with a small LOOCV score are preferred.

Fortunately, there is an amazing shortcut to calculating the LOOCV score.

Assume that the observed response values $Y_i$ are placed into a vector $\mathbf{Y}$, and the fitted values are placed in vector $\widehat{\mathbf{Y}}$. If there is a matrix $\mathbf{L}$ such that

$$\widehat{\mathbf{Y}} = \mathbf{L}\mathbf{Y}$$

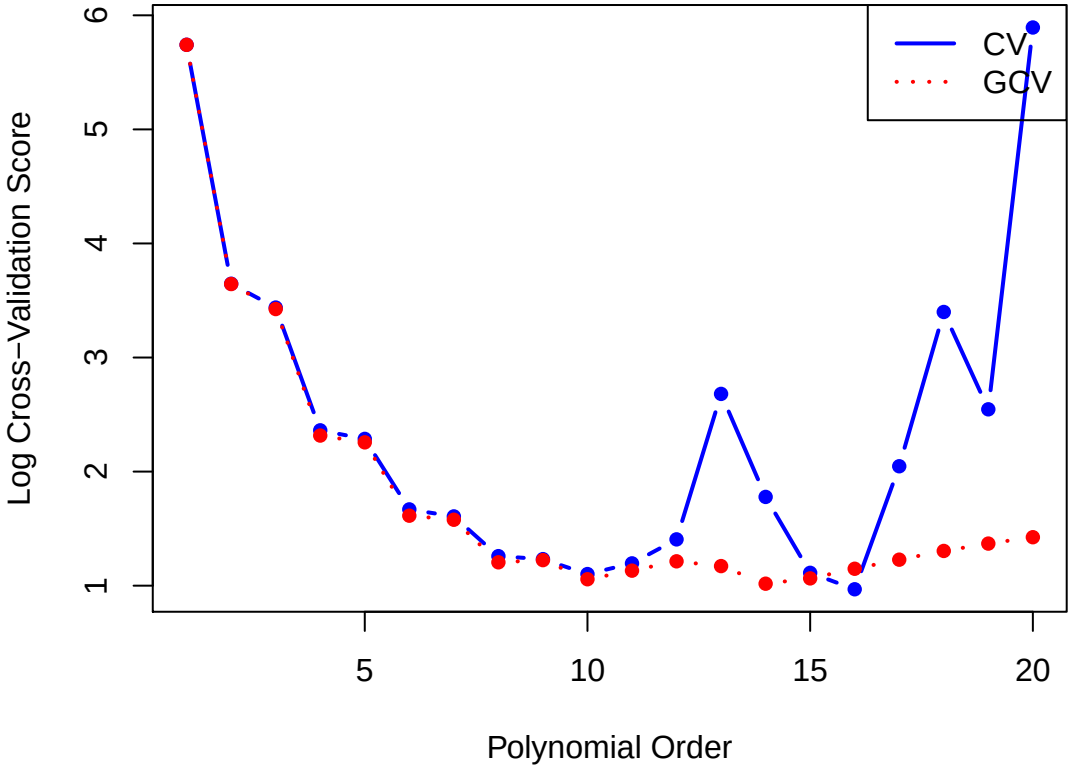
then

$$Y_i - \widehat{Y}_{(-i)} = \hat{\epsilon}_i \Big/ (1 - L_{ii})$$

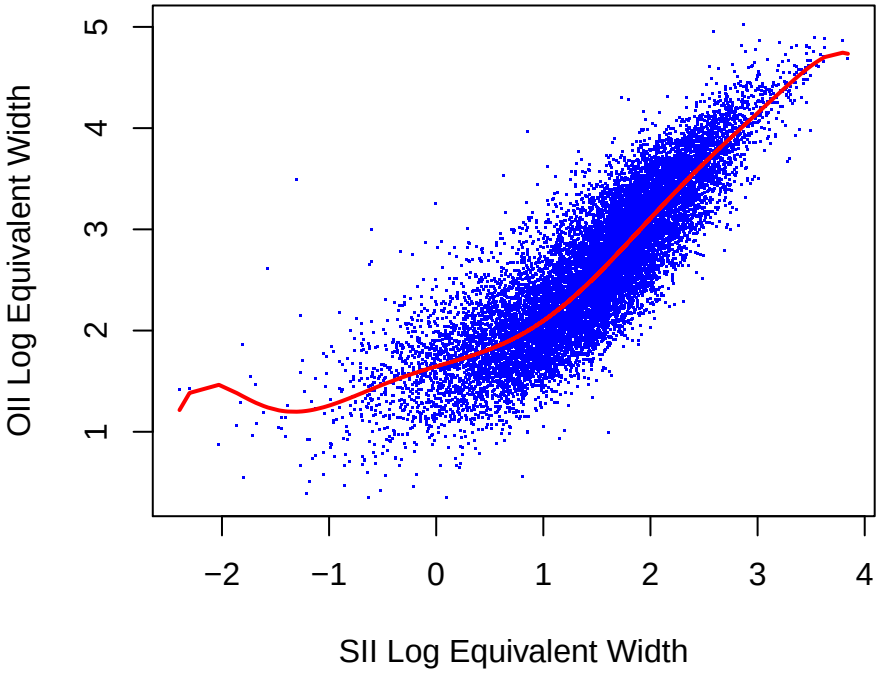
where $L_{ii}$ is the $i^{th}$ diagonal entry of $\mathbf{L}$. These are called the **leverages**.

The **generalized cross validation (GCV) score** is found by replacing $L_{ii}$ by the average diagonal entry of $\mathbf{L}$:

$$Y_i - \widehat{Y}_{(-i)} = \hat{\epsilon}_i \Big/ (1 - L_{ii}) \approx \hat{\epsilon}_i \Big/ (1 - \mathrm{tr}(\mathbf{L})/n)$$


CV
GCV
Log Cross−Validation Score
1
2
3
4
5
6
5
10
15
20
Polynomial Order

Including up to eighth power:

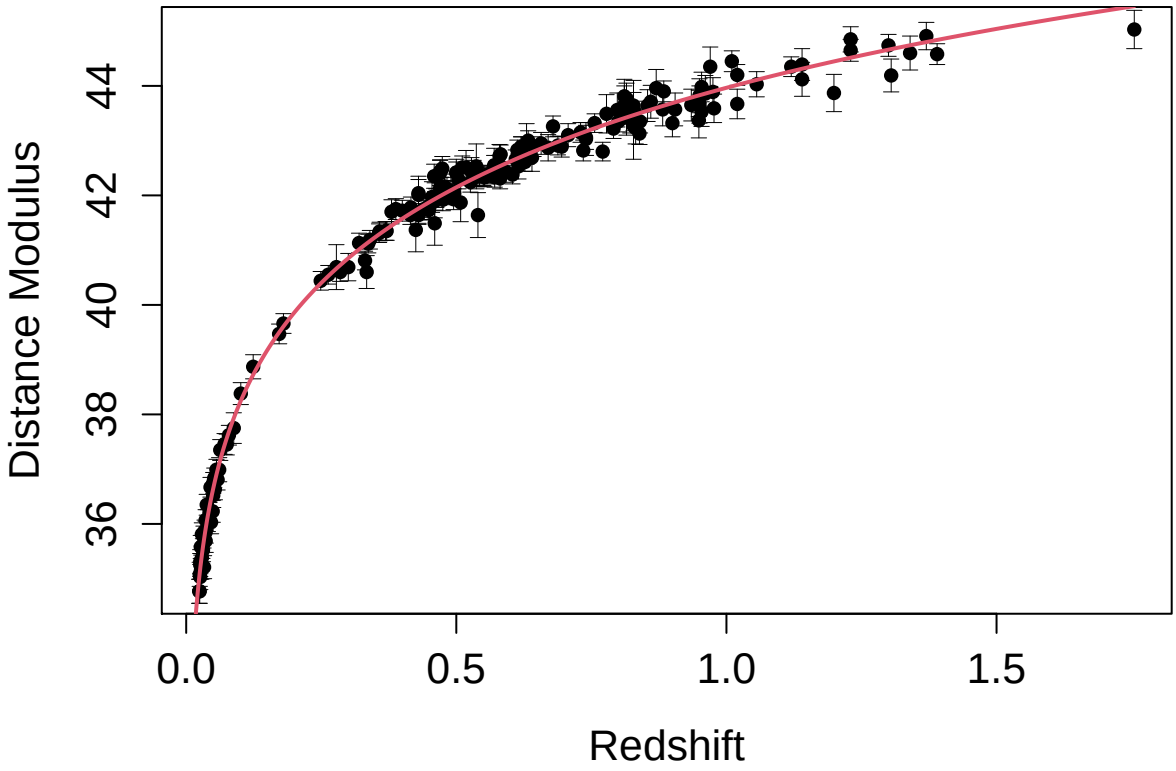


# Nonparametric Regression

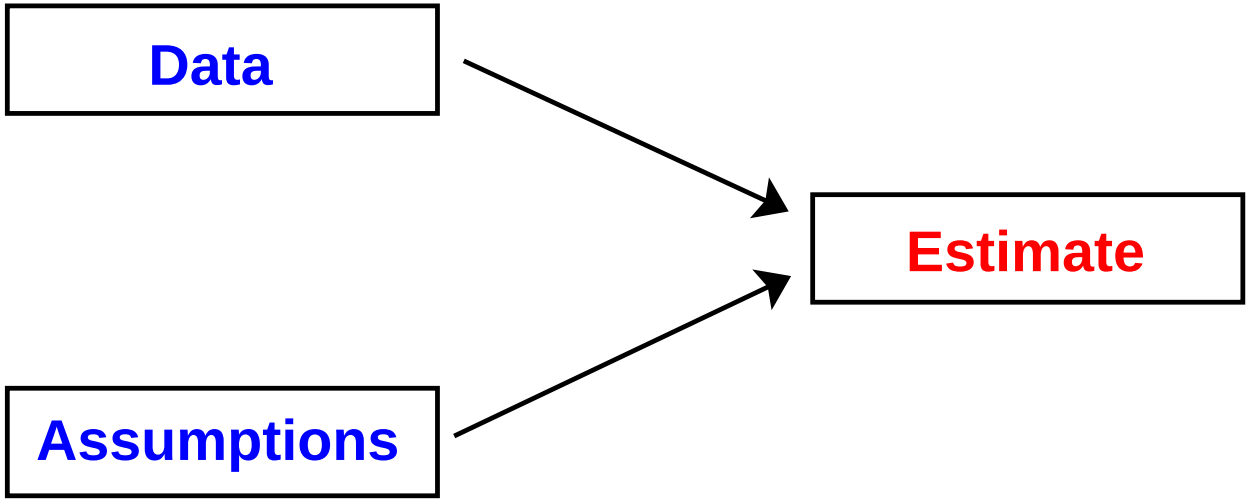
**Parametric** models have a fixed form for the relationship between the response and predictor(s):.

We saw at the start of the lecture that parametric models come in nonlinear forms as well, e.g.,

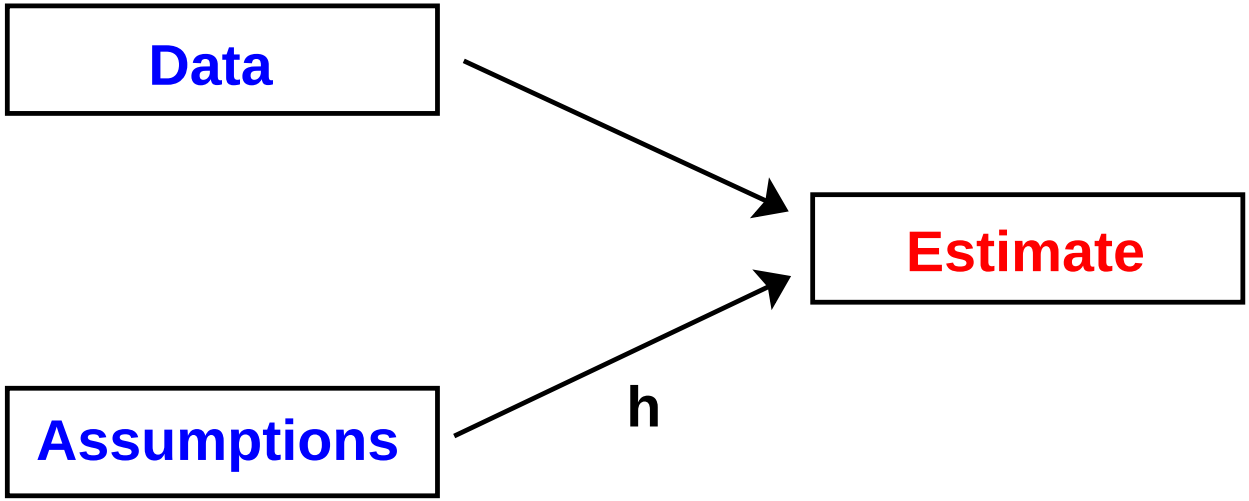
$$\text{Distance Modulus} = 5\log_{10}\left(\frac{c(1+z)}{H_0}\int_0^z \left(\Omega_m(1+u)^3 + (1-\Omega_m)\right)^{-1/2}\right) + 25$$



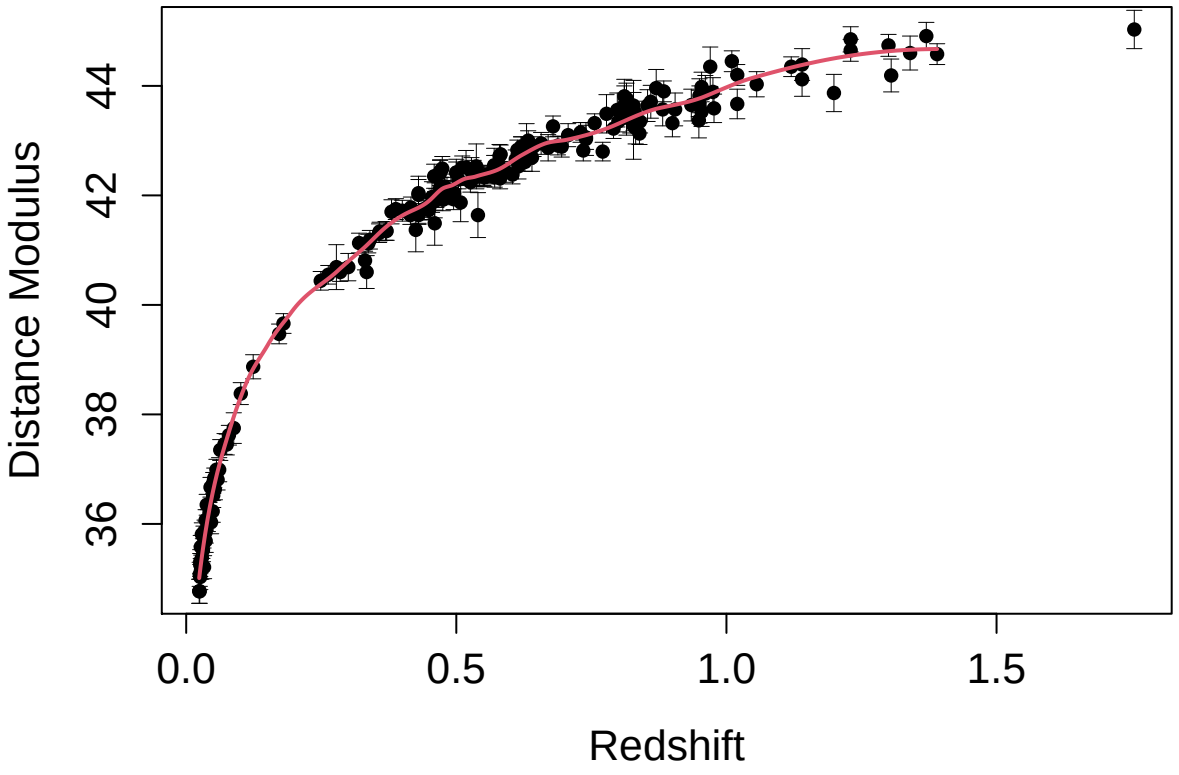

The case where $H_0 = 72.76$ and $\Omega_m = 0.34$.




In the **parametric case**, the contribution to the estimate from the assumptions is fixed.




In the **nonparametric case**, the influence of the assumptions is controlled by a **smoothing parameter** $h$. The value of $h$ can be chosen smaller with larger sample sizes.




A nonparametric estimate of the relationship.

## Local Linear Regression

The figure on the previous page shows a **nonparametric regression** or **nonparametric smooth** of the relationship.

There are many variants of nonparametric regression, but here we focus on **local linear regression**, a version of **local polynomial regression**. This approach has proven to be very powerful, and possesses many strong theoretical properties to justify its use.

Briefly stated, this procedure works by fitting a sequence of linear models: Each is fit not to the entire data set, but to only data within a **neighborhood** of a target point. The size of this neighborhood is the smoothing parameter: Large neighborhoods yield a large degree of smoothing, while small neighborhoods result in minimal smoothing.

Our model here is that we observe $(x_i, Y_i)$ for $i = 1, 2, \ldots, n$ and that

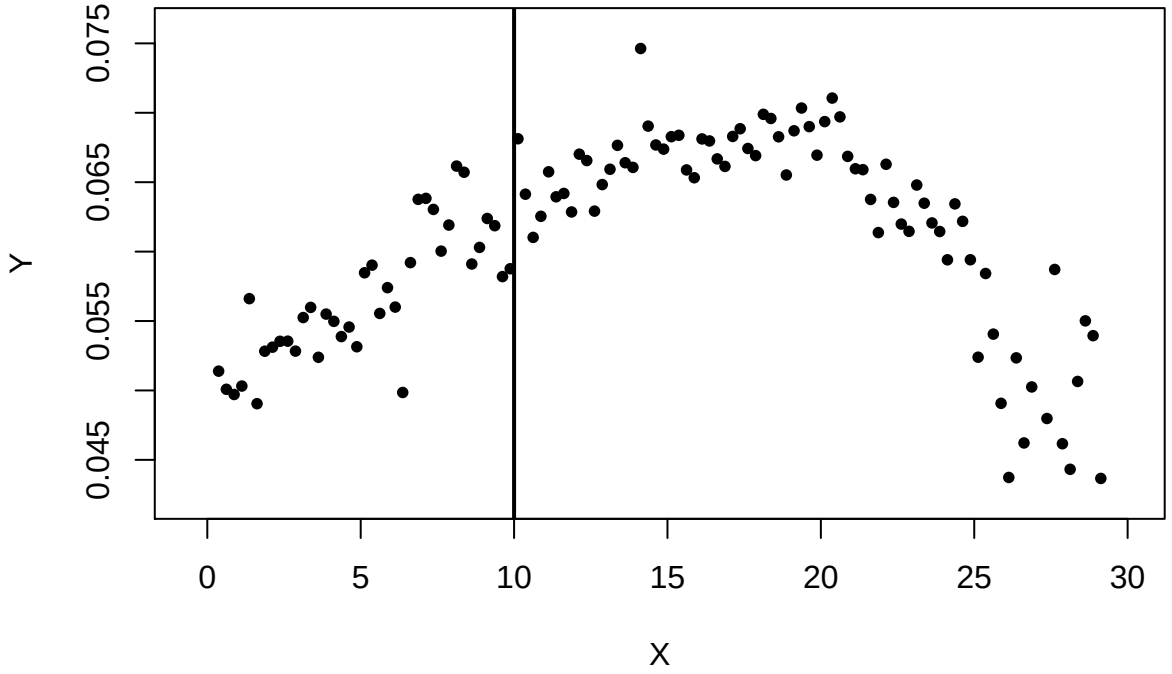
$$Y_i = f(x_i) + \epsilon_i$$

where the $\epsilon_i$ are iid with mean zero and variance $\sigma^2$. Assuming that the $\epsilon_i$ are normal will lead to some nice properties, but this development does not require that assumption.

In order to construct the local linear regression estimate of $f(\cdot)$, it is best to consider a sequence of steps **for each fixed $x_0$ at which $f(\cdot)$ will be estimated**.

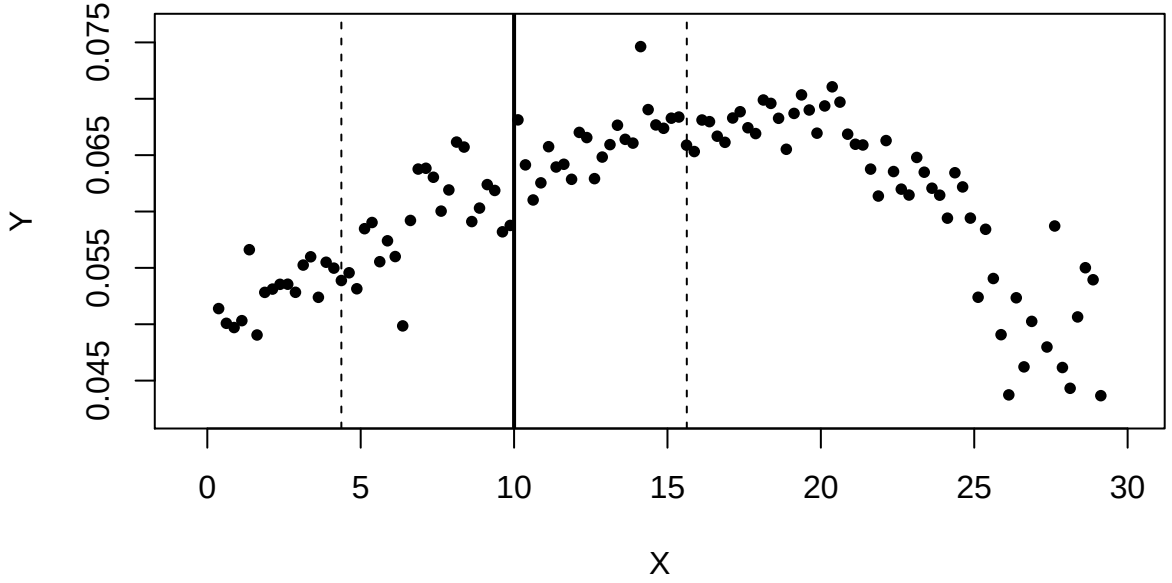
**Step One: Fix the target point $x_0$.**

Our objective is to estimate the regression function at $x_0$.



**Step Two: Create the neighborhood around $x_0$.**

A common way to choose the neighborhood size is to choose is large enough to capture proportion $\alpha$ of the data. This parameter $\alpha$ is often called the **span**. A typical choice is $\alpha \approx 0.5$.



**Step Three: Weight the data in the neighborhood.**

Values of $x$ which are close $x_0$ will receive a larger weight than those far from $x_0$.

Weighting is provided by a **kernel function**.

Denote by $w_i$ the weight placed on observation $i$. The default choice is the **tri-cube weight function**:

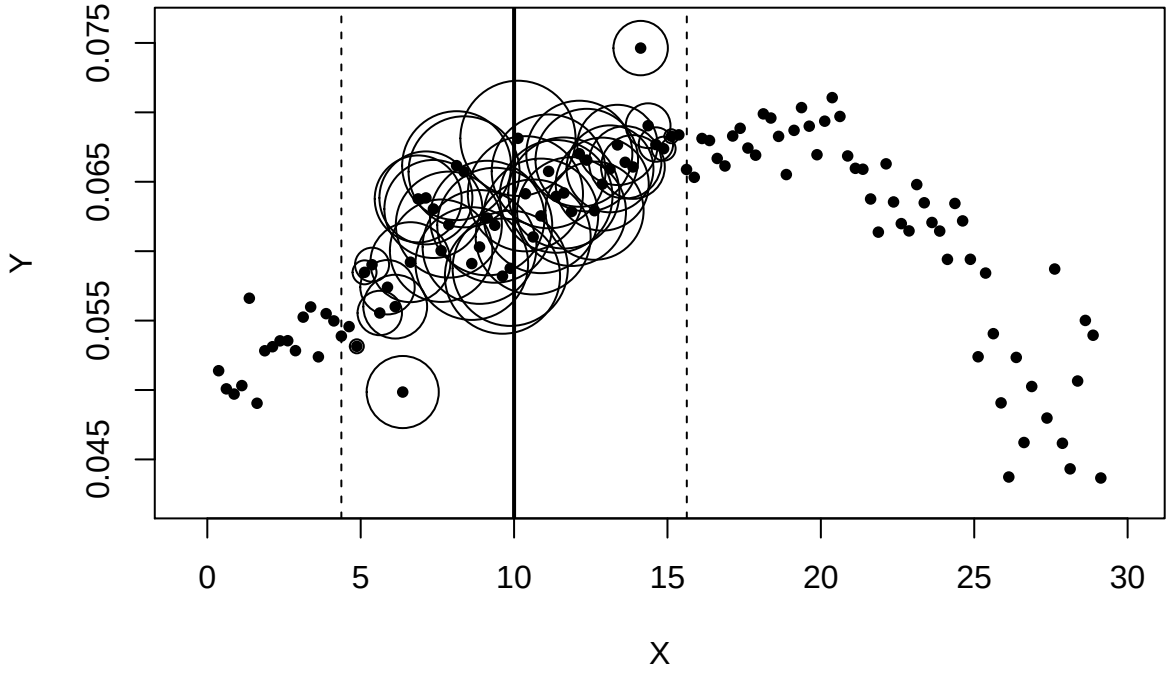
$$w_i = \begin{cases} \left(1 - \left|\frac{x_i - x_0}{\text{max dist}}\right|^3\right)^3, & \text{if } x_i \text{ in the neighborhood of } x_0 \\ 0, & \text{if } x_i \text{ is not in neighborhood of } x_0 \end{cases}$$

A **kernel** is any smooth function $K$ such that $K(x) \geq 0$ and

$$\int K(x)\,dx = 1, \quad \int xK(x)dx = 0 \quad \text{and} \quad \sigma_K^2 \equiv \int x^2 K(x)dx > 0.$$

Some commonly used kernels are the following:

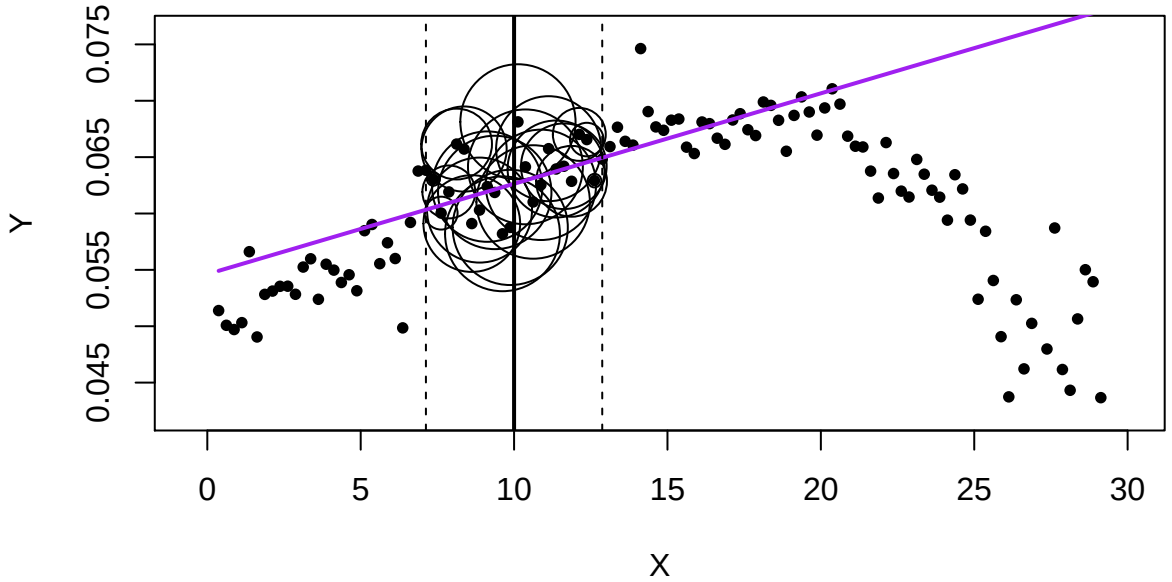
$$\begin{aligned}
\text{the boxcar kernel:} \quad & K(x) = I(|x| < 1), \\
\text{the Gaussian kernel:} \quad & K(x) = e^{-x^2/2}, \\
\text{the Epanechnikov kernel:} \quad & K(x) = (1 - x^2)I(|x| < 1) \\
\text{the tricube kernel:} \quad & K(x) = (1 - |x|^3)^3 I(|x| < 1).
\end{aligned}$$


Y
0.045
0.055
0.065
0.075
0
5
10
15
20
25
30
X

**Step Four: Fit the local regression line.**

This is done by finding $\beta_0$ and $\beta_1$ to minimize the weighted sum of squares

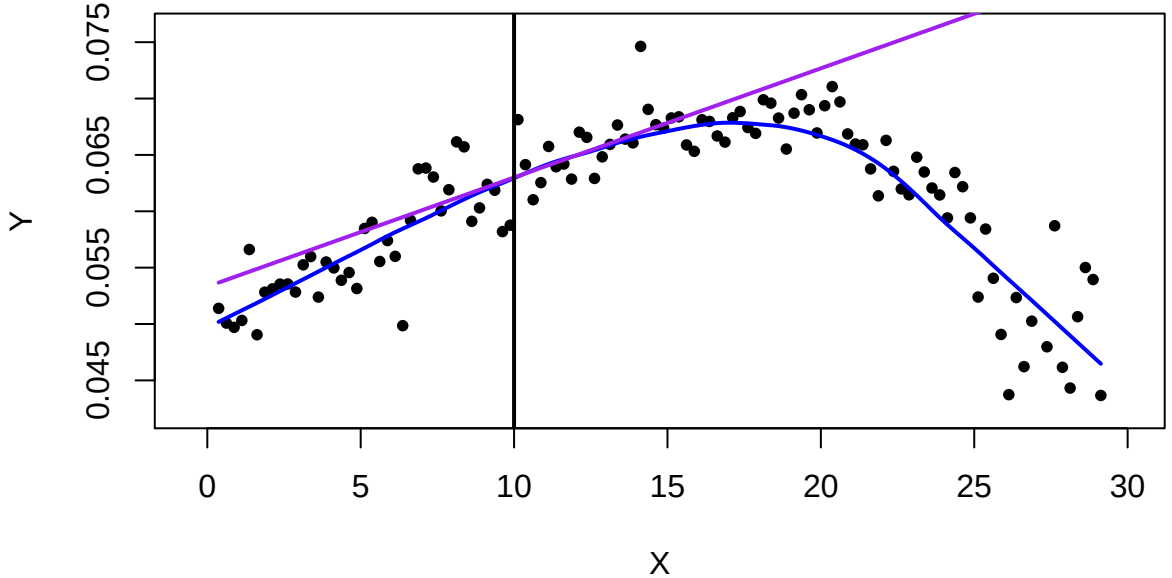
$$\sum_{i=1}^{n} w_i \left(y_i - (\beta_0 + \beta_1 x_i)\right)^2$$



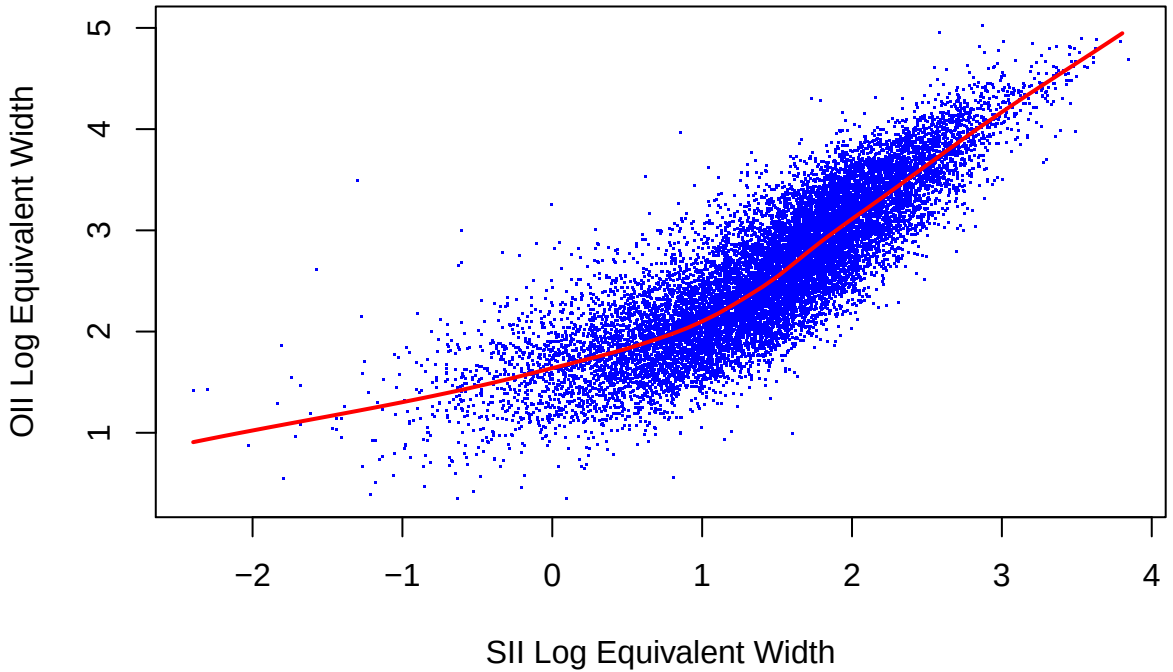
**Step Five: Estimate $f(x_0)$.**

This is done using the fitted regression line to estimate the regression function at $x_0$:

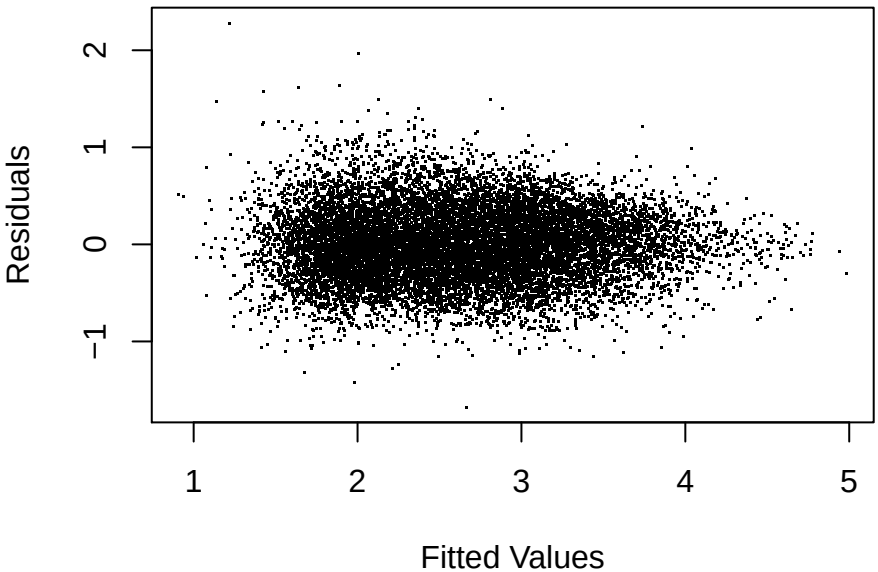
$$\widehat{f}(x_0) = \widehat{\beta}_0 + \widehat{\beta}_1 x_0$$



Let's return to the galaxy emission line data.



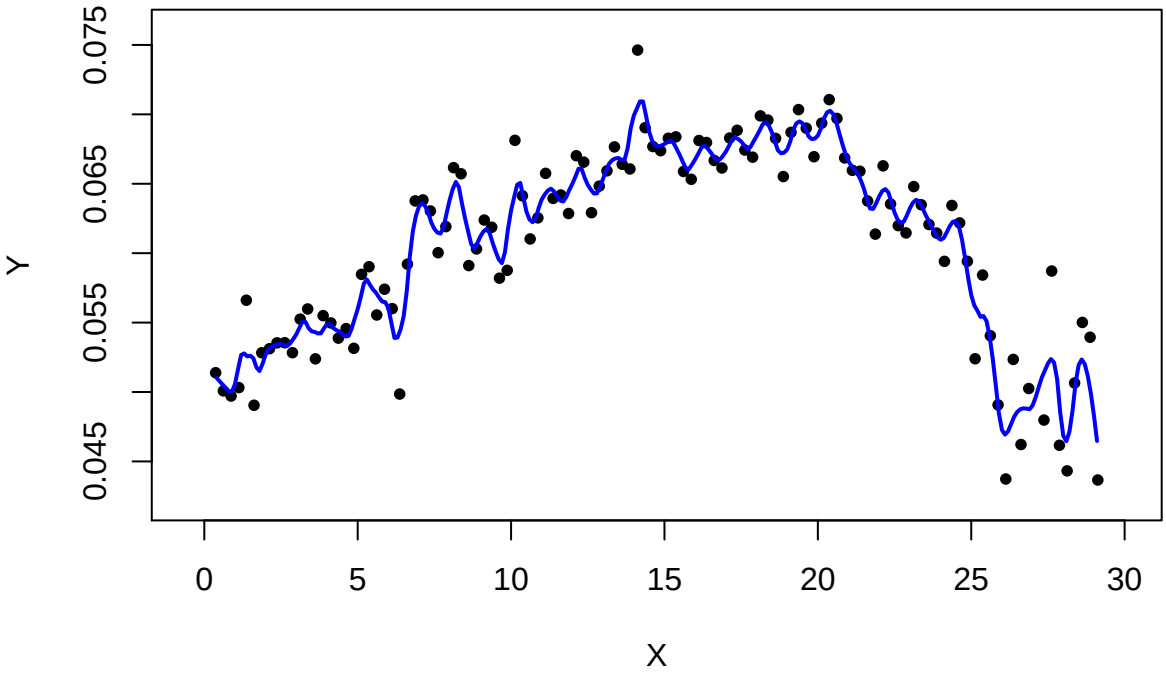
It still makes sense to look at the plot of residuals versus fitted values. The improvement in the fit is noticeable.



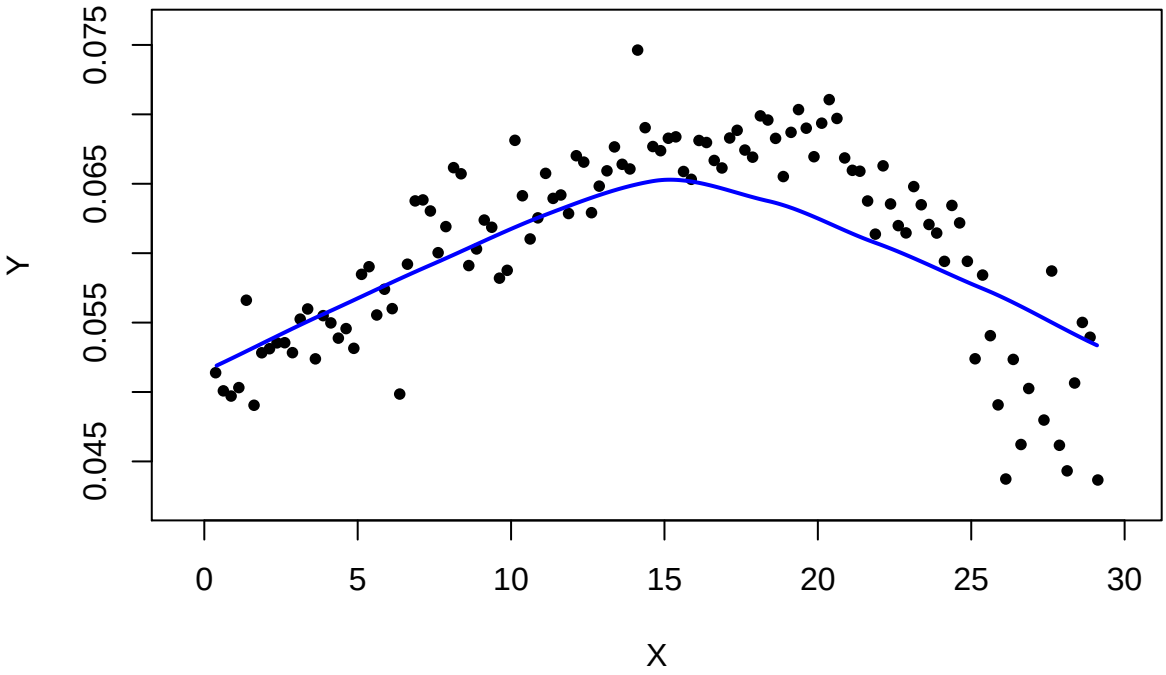
## Smoothing Parameter Selection

The smoothing parameter in nonparametric regression controls how "smooth" the resulting estimate is: A smaller value leads to a "rougher" estimate, a larger value gives a "smoother" estimate.

The smoothing parameter **cannot** be chosen by minimizing the sum of squared residuals ("least squares"). Doing so would lead to overfitting, since for small enough choice, the residuals could all be made close to zero.




Using span = 0.05. Clearly not enough smoothing.




Using span = 0.9. Too much smoothing, missing important features.

As before, a common strategy for setting smoothing parameters is to use cross-validation.

The LOOCV score can be calculated for different values of the smoothing parameter:

$$\text{LOOCV}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_{(-i)}\right)^2$$

and the choice of $h$ which minimizes the score is optimal. GCV is also often used.

In the galaxy emission line example above, the span was chosen to be 0.27 based on GCV.

## A Bit of Theory

Estimators are assessed via their **integrated mean squared error**:

$$E\Big[\int \big(\widehat{f}(x) - f(x)\big)^2 \, dx\Big] = \text{bias}^2(\widehat{f}) + \text{Var}(\widehat{f})$$

The expectation accounts for randomness in the estimator, i.e., averaging over all hypothetical samples.

This illustrates the crucial **bias/variance decomposition**.

Our first attempt at nonparametric regression may be **local averaging** of the following form:

$$\hat{f}(x) = \text{mean}\left\{ Y_i : \; x - \frac{h}{2} < X_i < x - \frac{h}{2} \right\}.$$

This is like **binning** but removes the arbitrary boundaries.

The local average estimator can be written as

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} Y_i \; K\left(\frac{x - X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}$$

where

$$K(x) = \begin{cases} 1 & |x| < 1/2 \\ 0 & \text{otherwise.} \end{cases}$$

Can improve this by using a function $K$ which is **smoother**.

Choose $K$ to be a smooth **kernel function**, as defined above.

This is called **kernel regression**.

Both kernel regression and local linear regression have the same (approximate) variance.

But, the kernel estimator has bias

$$h^2 \left( \frac{1}{2} f''(x) + \frac{f'(x)g'(x)}{g(x)} \right) \int u^2 K(u) du$$

whereas the local linear estimator only has bias

$$h^2 \frac{1}{2} f''(x) \int u^2 K(u) du$$

Also, at the boundary points, the kernel estimator has asymptotic bias of $O(h)$ while the local linear estimator has bias $O(h^2)$.

## An Aside: Linear Smoothers

The local polynomial regression estimate is

$$\hat{f}_n(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

where $\ell(x)^T = (\ell_1(x), \ldots, \ell_n(x))$,

$$\ell(x)^T = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x,$$

$e_1 = (1, 0, \ldots, 0)^T$ and $X_x$ and $W_x$ are defined by

$$X_x = \begin{pmatrix} 1 & X_1 - x & \cdots & \frac{(X_1-x)^p}{p!} \\ 1 & X_2 - x & \cdots & \frac{(X_2-x)^p}{p!} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n - x & \cdots & \frac{(X_n-x)^p}{p!} \end{pmatrix} \quad W_x = \begin{pmatrix} K\left(\frac{x-X_1}{h}\right) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & K\left(\frac{x-X_n}{h}\right) \end{pmatrix}.$$

This means that local polynomial regression is a **linear smoother**, i.e.,

$$\widehat{\mathbf{Y}} = \mathbf{L}\mathbf{Y}$$

where $\mathbf{L}$ is the **smoothing matrix**:

$$\mathbf{L} = \begin{pmatrix} \ell_1(X_1) & \ell_2(X_1) & \cdots & \ell_n(X_1) \\ \ell_1(X_2) & \ell_2(X_2) & \cdots & \ell_n(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ \ell_1(X_n) & \ell_2(X_n) & \cdots & \ell_n(X_n) \end{pmatrix}.$$

The **effective degrees of freedom** is:

$$\nu = \text{trace}(\mathbf{L}) = \sum_{i=1}^{n} L_{ii}.$$

Also greatly eases the calculation of leave-one-out residuals. Recall:

$$Y_i - \widehat{Y}_{(-i)} = \widehat{\epsilon}_i \Big/ (1 - L_{ii})$$

# The Curse of Dimensionality

Despite the promise of nonparametric regression, they suffer from a serious drawback: The amount of data required to fit these models well increases exponentially with the dimension of the data.

**What is the "dimension" of data?**

Standard data come to us in the form of numbers, but a single **observation** is often best thought of as a vector.

Examples include images, spectra, photometry, light curves, etc.

If there are $p$ predictors being used in a model for a response variable, we would say that we are using a "$p$-dimensional predictor vector." In some applications, $p$ can be very large.

Nonparametric regression is possible, in theory. For instance, if $p = 2$, a neighborhood can be thought of as a square centered on the target value. Local linear regression involves fitting a plane within this square.

But, as the dimension increases, the available observations become more "spread out." Neighborhoods must be made larger in order to compensate and achieve estimates with acceptable variance.

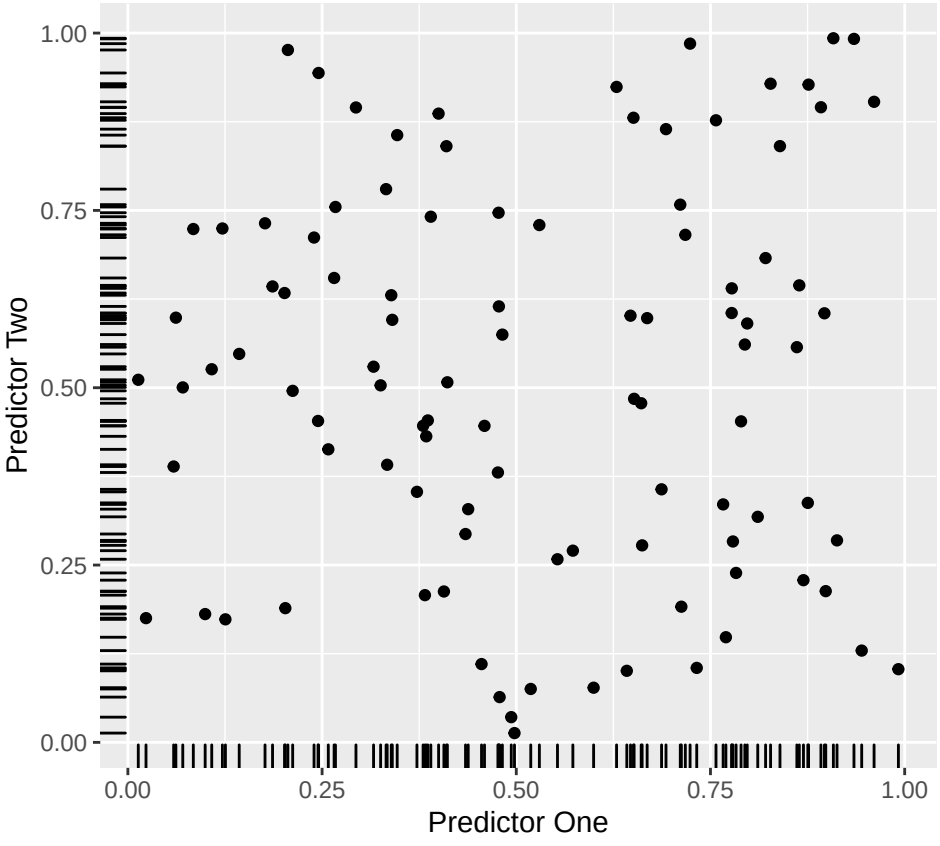
Choosing a large neighborhood takes away a key feature of using nonparametric approaches: We want the estimate to be flexible, and not be based on too much "smoothing."

Consider the plot on the following slide.

Here, the sample size is 100. Each observation is two-dimensional, i.e., imagine these are the two predictors. These data are simulated from uniform distributions.

The "marks" shown in each margin depict the **marginal distribution** for each predictor. Note that in the margins, the data appear to be quite closely spaced. If we were to fit a nonparametric regression using just one predictor, neighborhoods could be chosen small.

But, in two dimensions, large gaps appear in the distribution of the observations. Neighborhoods will have to be chosen larger in order to capture enough data.


1.00
0.75
0.50
0.25
0.00
Predictor Two
0.00
0.25
0.50
0.75
1.00
Predictor One

One approach to dealing with the curse of dimensionality is to utilize techniques of **dimension reduction** to map high-dimensional data into low-dimensional representations.

**Principal components analysis (PCA)** is a classic method to dimension reduction, but more-sophisticated **nonlinear** approaches such as **isomap**, **local linear embeddings**, and **diffusion map** are available.

# Additive Models

Another approach to dealing with the curse of dimensionality is to fit models which are nonparametric, but are **additive** in the predictors.

Consider a case with $p = 3$ predictors. A **fully nonparametric model** would be of the form

$$Y = f(X_1, X_2, X_3) + \epsilon,$$

i.e., the function $f()$ could take any form on the three-dimensional predictor space. This model is very flexible, but difficult to fit well due to the curse of dimensionality.

An **additive nonparametric model** would assume

$$Y = \beta_0 + f_1(X_1) + f_2(X_2) + f_3(X_3) + \epsilon$$

where each of the $f_i()$ are estimated **nonparametrically**.

The general estimation strategy is called **backfitting**.

In this process, each $f_k$ is estimated nonparametrically, in a rotation, and the process is repeated until there is convergence.

When estimating $f_k(\cdot)$, the other $f_j(\cdot)$ are held fixed at their current best estimates, and we set up a one-dimensional nonparametric estimation problem, on which one could use either local linear regression, smoothing splines, or other approach.

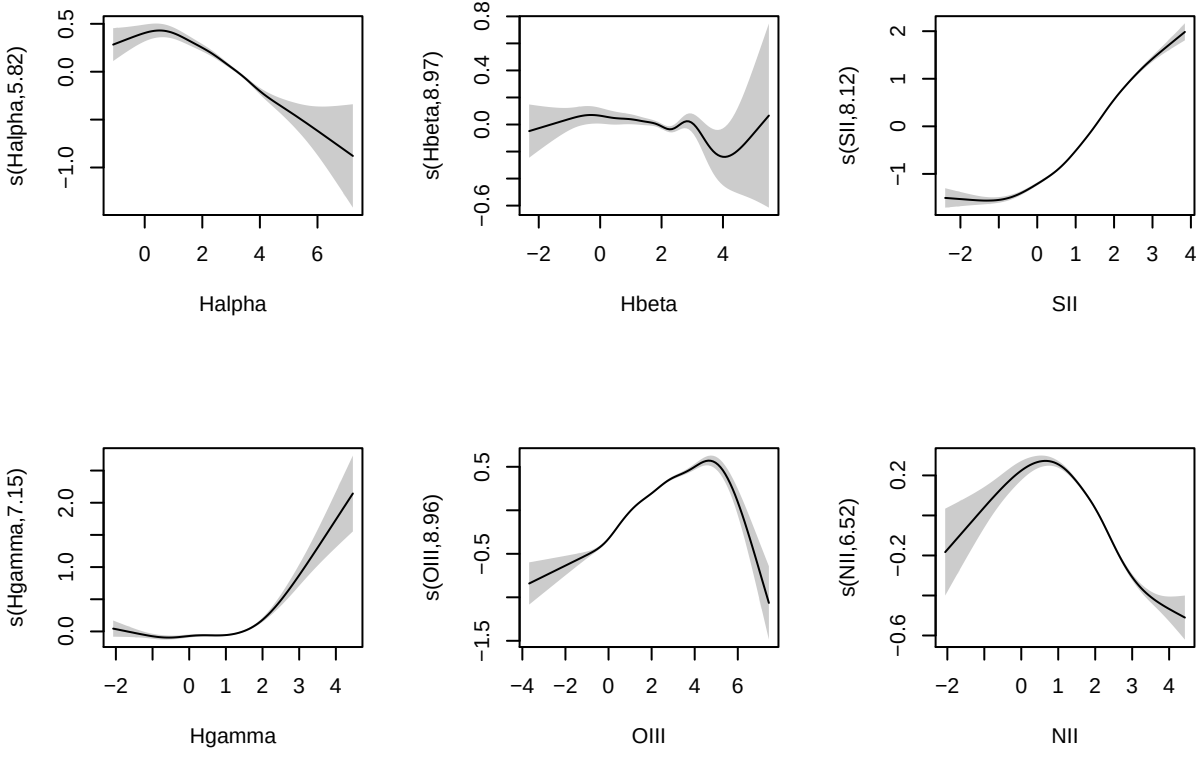
So, when estimating $f_k(\cdot)$, the response is taken to be

$$Y_i - \left[\sum_{j \neq k} \widehat{f}_j(x_{ij})\right]$$

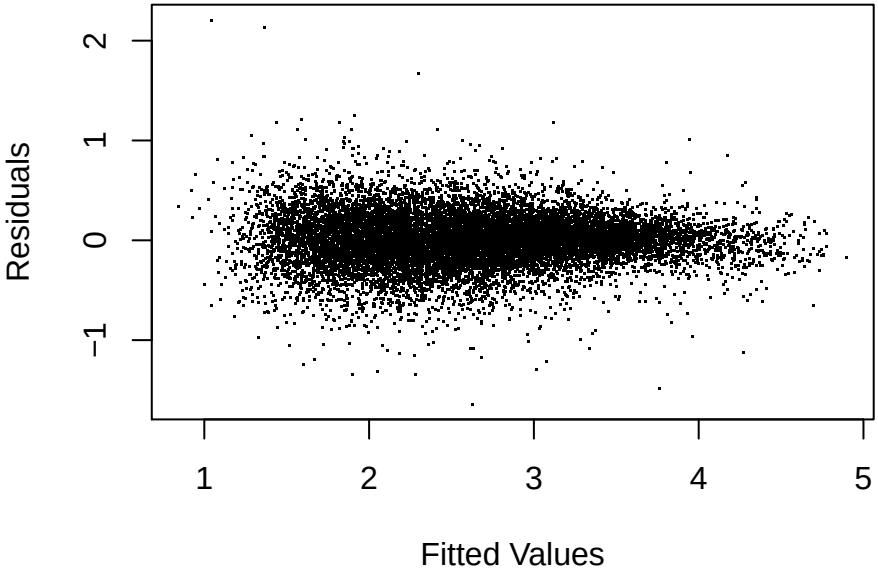
## Back to the SDSS Data

We can now fit an additive model relating the OII line strength to the other six lines.

The estimates of the functions $f_i$ can be viewed.


s(Halpha,5.82)
−1.0
0.0
0.5
0
2
4
6
Halpha
s(Hbeta,8.97)
−0.6
0.0
0.4
0.8
−2
0
2
4
Hbeta
s(SII,8.12)
−1
0
1
2
−2
0
1
2
3
4
SII
s(Hgamma,7.15)
0.0
1.0
2.0
−2
0
1
2
3
4
Hgamma
s(OIII,8.96)
−1.5
−0.5
0.5
−4
−2
0
2
4
6
OIII
s(NII,6.52)
−0.6
−0.2
0.2
−2
0
1
2
3
4
NII

It again is important to look at the plot of residuals versus fitted values. There is no evidence of a lack of fit.



## Projection Pursuit Regression

Recall our additive nonparametric regression model:

$$Y = \beta_0 + \sum_{j=1}^{p} f_j(x_j) + \epsilon$$

A generalization of this model is the **projection pursuit regression** model, which can be written as

$$Y = \beta_0 + \sum_{k=1}^{M} \beta_k f_k\left(\boldsymbol{\alpha}_k^T \mathbf{x}\right) + \epsilon$$

where each of the $\boldsymbol{\alpha}_k$ are a vector of length $p$. These $\boldsymbol{\alpha}_k$ are the **projection direction vectors**.

The functions $f_k$, called the **ridge functions**, are estimated nonparametrically. These functions are scaled to have mean zero and variance one when applied to the observed sample.

The projection pursuit model takes linear combinations of the predictors, and then fits an additive model in these linear combinations. One can think of the $\boldsymbol{\alpha}_k^T \mathbf{x}$ as being "new predictors" which are being utilized in an additive model.

Projection pursuit is related to basic **neural network models**.

$$Y = \beta_0 + \sum_{k=1}^{M} \beta_k \phi(\alpha_{0k} + \boldsymbol{\alpha}_k^T \mathbf{x}) + \epsilon$$

Here, $\phi$ is a given function (the **activation function**), and not estimated from data.

# Conclusion

Motivation

Underlying Concepts

Nonparametric Approaches

Some Theory

The Curse of Dimensionality

Additive Models

## A Basis Method: Smoothing Splines

Another nonparametric approach to regression is the **penalized spline** or **smoothing spline**.

This approach starts with a natural optimization problem: Find the twice differentiable function $f(\cdot)$ such that

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_a^b \left[f^{(2)}(x)\right]^2 dx$$

where $f^{(2)}(x)$ indicates the second derivative of $f$ evaluated at $x$ and $\lambda > 0$ is the smoothing parameter.

Typically, $a = \min\{x_i\}$ and $b = \max\{x_i\}$.

Note that the **penalty term**

$$\int_a^b \left[f^{(2)}(x)\right]^2$$

will be large if the function is "wiggly." Of course, if $f(x) = a + bx$, then this penalty equals zero.

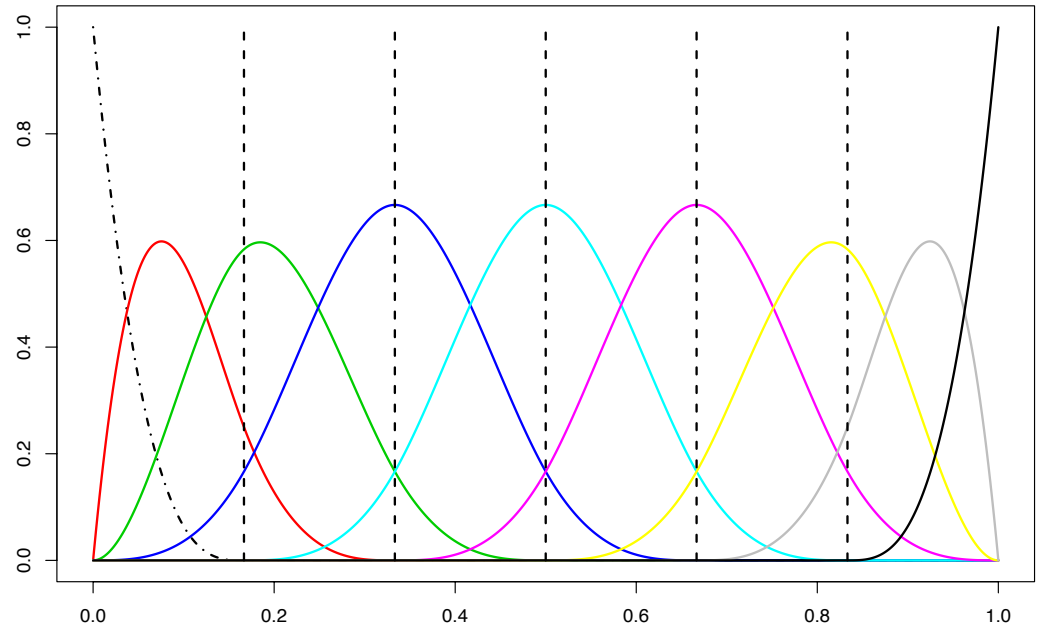
Large values of $\lambda$ lead to smooth functions $f()$.

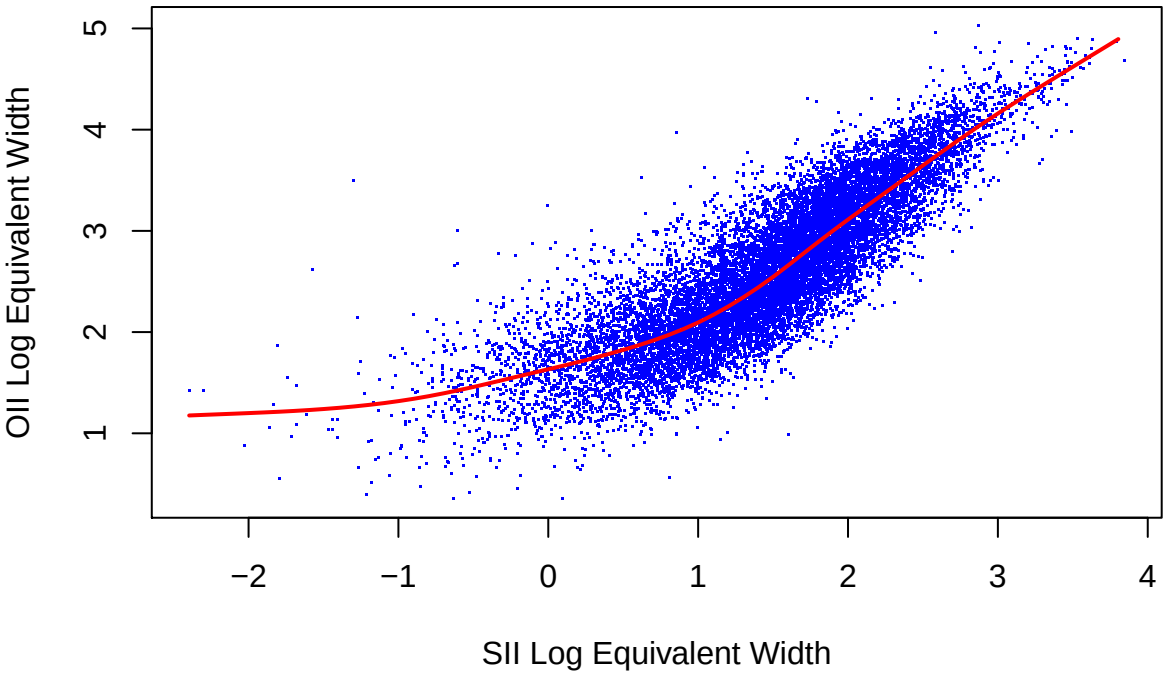
As before, $\lambda$ can be chosen via cross-validation.

This may appear to set up a difficult optimization problem, but, in fact, the search for the optimal $f(\cdot)$ can be transformed into constructing $\widehat{f}(\cdot)$ as the linear combination of a specially formed **basis** of functions.

**Comment:** The number of basis functions utilized is equal to the number of **knots** plus four. For computational reasons, the number of knots is usually chosen much smaller than the number of data points.

The figure below shows a simple example for the case where there are five knots (shown as the vertical lines).




OII Log Equivalent Width
1
2
3
4
5
−2
−1
0
1
2
3
4
SII Log Equivalent Width