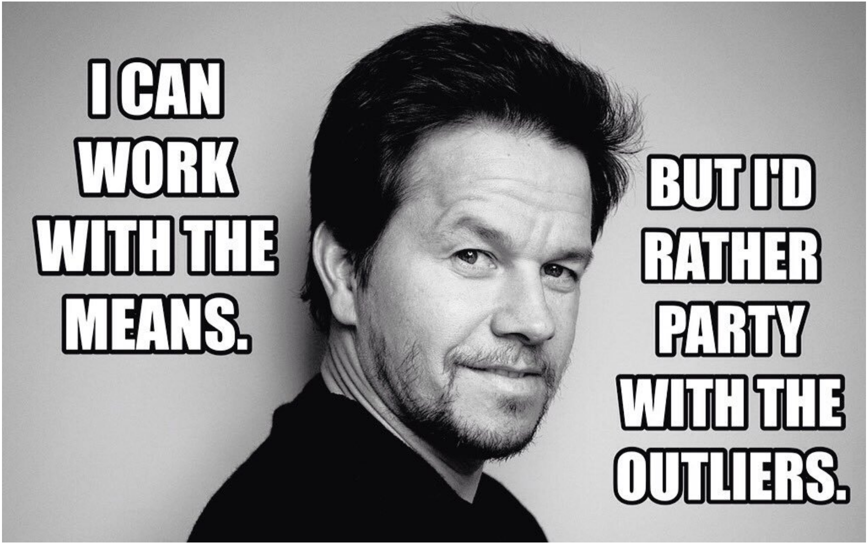
STAT406- Methods of Statistical Learning Lecture 9

Matias Salibian-Barrera

UBC - Sep / Dec 2017


I CAN WORK WITH THE MEANS.
BUT I'D RATHER PARTY WITH THE OUTLIERS.

Kernel smoothers

- We are interested in estimating

$$f(x) = E\left(Y \middle| X = x\right)$$

- Given a sample (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)

$$\hat{f}(x) = \text{average}\Big\{ y_j : x_j = x \Big\}$$

$$\hat{f}(x) = \text{average}\Big\{ y_j : x_j \text{ is close to } x \Big\}$$

Kernel smoothers

- More formally

$$\hat{f}(x) \;=\; \text{average}\left\{ \, y_j \, : \, \left| x_j - x \right| \le h \, \right\}$$

$$\hat{f}(x) \;=\; \frac{1}{n_x} \sum_{i \,:\, |x_i - x| \le h} y_i$$

$$\hat{f}(x) \;=\; \frac{\sum_{i \,:\, |x_i - x| \le h} y_i}{\sum_{i \,:\, |x_i - x| \le h} 1}$$

Kernel smoothers

- More formally

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K(x_i, x, h)\, y_i}{\sum_{j=1}^{n} K(x_j, x, h)}$$

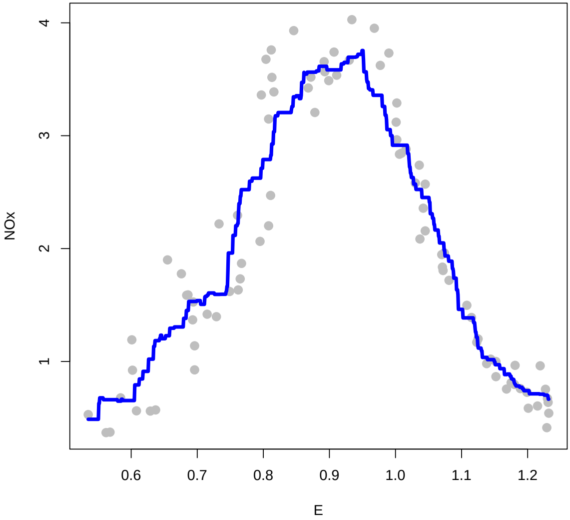
where

$$K(x_j, x, h) = W\left(\frac{x_j - x}{h}\right)$$

and

$$W(t) = \begin{cases} 1 & \text{if } |t| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Kernel smoothers - h = 0.07



Kernel smoothers

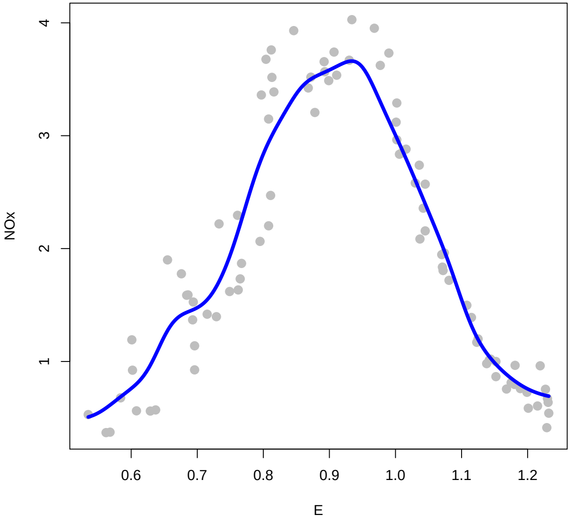
- Discontinuities come from $W(t)$
- Use a smooth kernel

$$K(x_j, x, h) \ = \ W\left(\frac{x_j - x}{h}\right)$$

with

$$W(t) \ = \ \begin{cases} 1 - t^2 & \text{if } |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Kernel smoothers - h = 0.03



Kernel smoothers

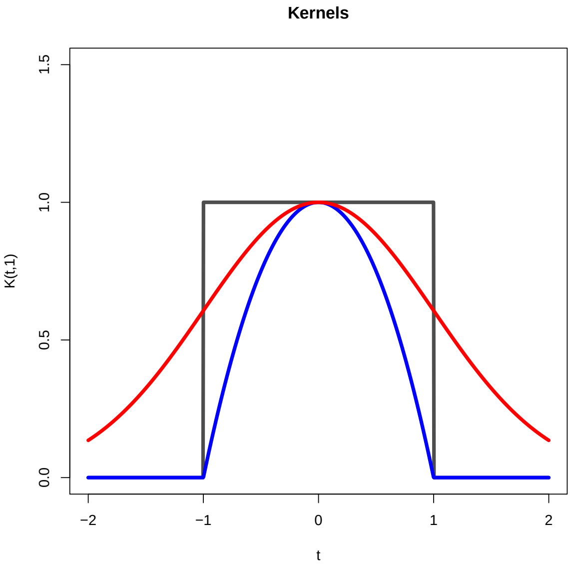
- Other kernels...

$$K(x_j, x, h) \; = \; W\left(\frac{x_j - x}{h}\right)$$

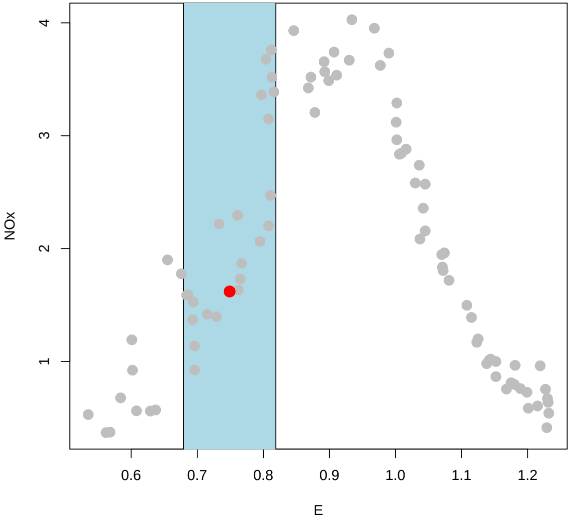
with

$$W(t) \; = \; \phi(t) \; \propto \; \exp\left(-t^2/2\right)$$

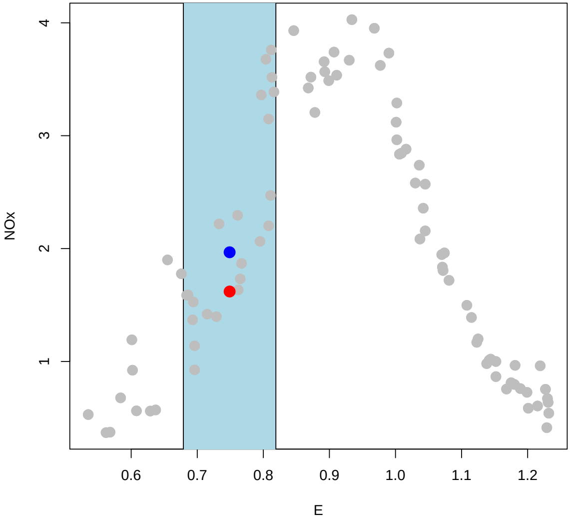
Kernel smoother in action



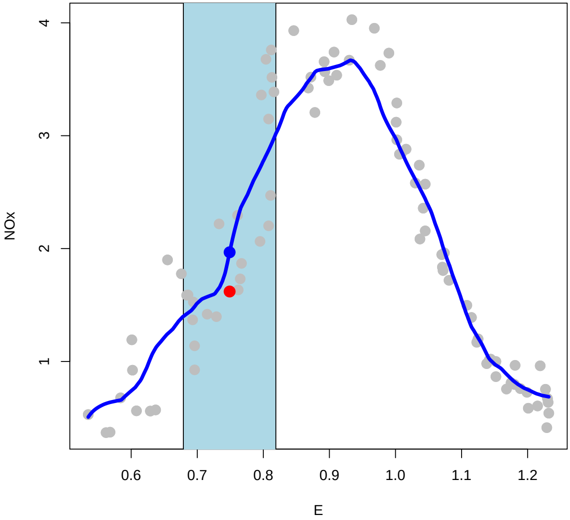
Kernel smoother in action



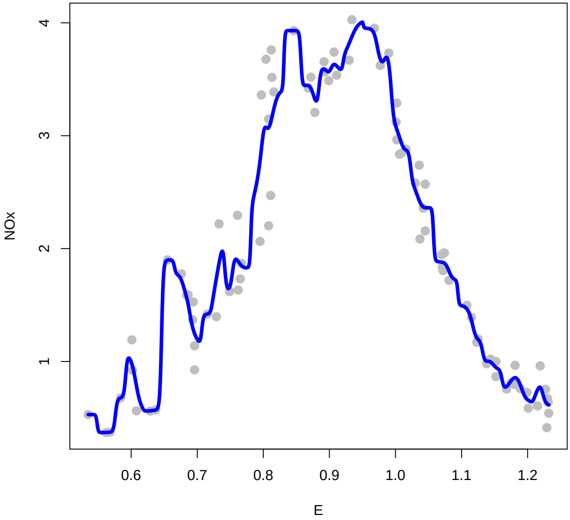
Kernel smoother in action



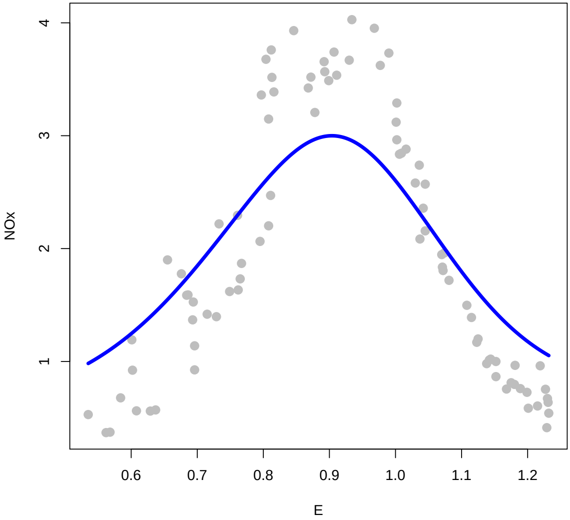
Kernel smoother in action



Small bandwidth



Larger bandwidth



Kernel smoothers

- Note that the locally weighted average

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K(x_i, x, h)\, y_i}{\sum_{j=1}^{n} K(x_j, x, h)}$$

solves

$$\hat{f}(x) = \arg\min_{\mu} \sum_{i=1}^{n} K(x_i, x, h)\, (y_i - \mu)^2$$

- "Local constant" fit

Kernel smoothers

- A Taylor expansion suggests

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + R$$

for small $|x - x_0|$

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) = \beta_0(x_0) + \beta_1(x_0)\, x$$

a "local" line

- "Local linear regression"

Kernel smoothers

- At each point x we find

$$\hat{\beta}(x) = \arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} K_h(x_i, x) \left(y_i - \beta_0 - \beta_1 x_i \right)^2$$

  which is just weighted least squares

- There is a closed form expression for $\hat{\beta}(x)$

Kernel smoothers

- We have that $\hat{\beta}(x)$ is

$$\hat{\beta}(x) = \left(\mathbf{X}' \, \mathbf{W}_x \, \mathbf{X}\right)^{-1} \mathbf{X}' \, \mathbf{W}_x \, \mathbf{Y}$$

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

$$\mathbf{W}_x = \text{diag} \, \left(K_h(x_1, x), \ldots, K_h(x_n, x) \right)$$

Kernel smoothers

- An alternative representation, $\hat{\beta}(x)$ solves

$$\arg\min_{\beta_0,\beta_1} \sum_{i=1}^{n} K_h(x_i, x) \left(y_i - \beta_0 - \beta_1 \, (x_i - x)\right)^2$$

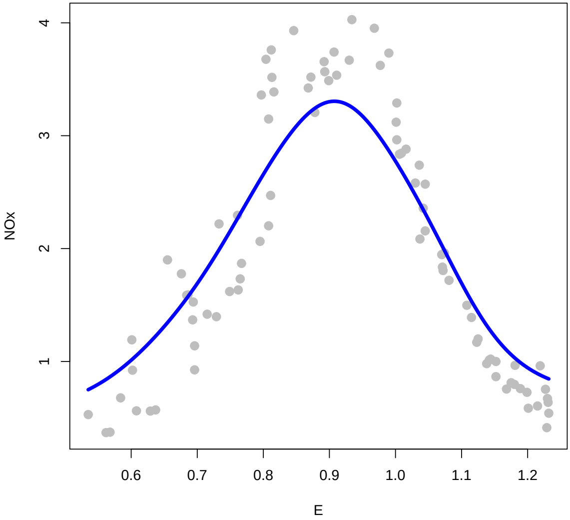
then the fit at x is $\hat{f}(x) = \hat{\beta}_0(x)$

- Note that

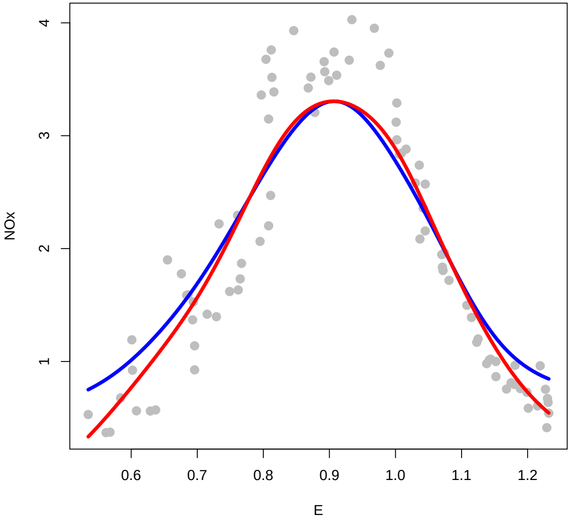
$$\hat{f}(x) = \mathbf{e}_1' \left(\tilde{\mathbf{X}}' \, \mathbf{W}_x \, \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}' \, \mathbf{W}_x \, \mathbf{Y}$$

- Local linear fits work better at the boundaries

Local mean - h = 0.07



Loc. mean & loc. linear - h = 0.07



Kernel smoothers

- These fits are linear
- In other words:

$$\hat{f}(x) = \ell(x)' \, \mathbf{Y}$$

hence, if Y_i are independent with $\mathrm{Var}(Y_i) = \sigma^2$, $1 \le i \le n$, then:

$$\mathrm{Var}(\hat{f}(x)) = \|\ell(x)\|^2 \, \sigma^2$$

Kernel smoothers

- We can also consider quadratic fits

$$\hat{\beta}(x) \; = \; \arg\min_{\beta} \; \sum_{i=1}^{n} K(x_i, x, h) \, (y_i - \beta' \mathbf{Z}_i)^2$$

where

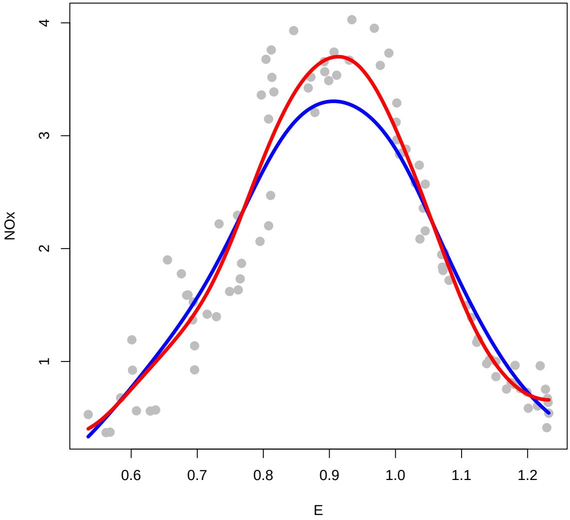
$$\mathbf{Z}_i \; = \; \left(\begin{array}{c} 1 \\ x_i - x \\ (x_i - x)^2 \end{array} \right)$$

and again

$$\hat{f}(x) \; = \; \hat{\beta}(x)_1 \; = \; \mathbf{e}_1' \hat{\beta}(x) \qquad \text{(the intercept)}$$

- Better fit for “valleys” and “peaks”

Local lin. & quad. - h = 0.07



Kernel smoothers

- We can use higher degrees:

$$\hat{\beta}(x) = \arg\min_{\beta} \sum_{i=1}^{n} K(x_i, x, h)\,(y_i - \beta' \mathbf{Z}_i)^2$$

where

$$\mathbf{Z}_i = \begin{pmatrix} 1 \\ x_i - x \\ \vdots \\ (x_i - x)^p \end{pmatrix}$$

$$\hat{\beta}(x) = (\mathbf{Z}' \mathbf{W} \mathbf{Z})^{-1} \mathbf{Z}' \mathbf{W} \mathbf{Y}$$

$$\hat{f}(x) = \mathbf{e}_1' \hat{\beta}(x) = \ell(x)' \mathbf{Y}$$

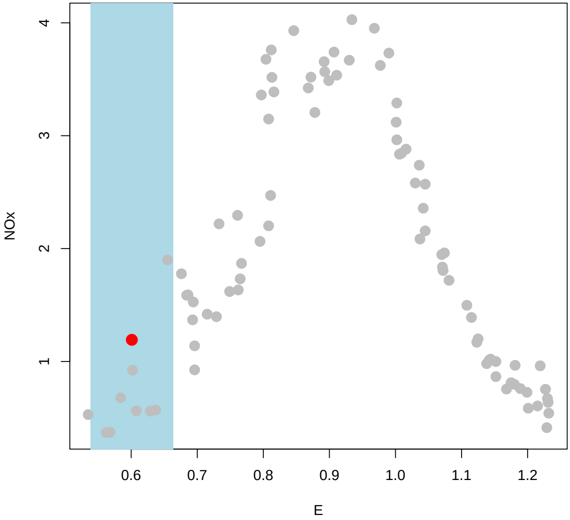
Varying bandwidths...

- Nearest neighbours?
- Use a different h for each x_0

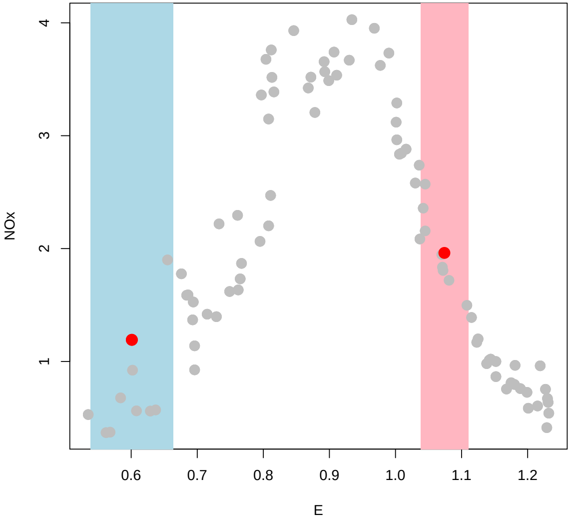
$$\min_{\beta} \sum_{i=1}^{n} K_{h(x_0)}(x_i, x_0) \left(y_i - \beta_0 - \beta_1 (x_i - x_0) \right)^2$$

- Make the window wider to avoid "empty" or "sparse" windows
- For example, we want 10% of our sample in **every window**

Varying window, $\alpha = 0.10$



Varying window, $\alpha = 0.10$



Varying bandwidths...

- More in general: let $\alpha \in (0, 1]$ and choose

$$h(x_0) \; = \; d_{([n\,\alpha])}$$

where

$$d_j \; = \; \left| x_j - x_0 \right| \qquad j = 1, \ldots, n$$

- Always have α 100 % of the points in each window
- This, again, is a linear smoother:

$$\hat{f}(x_0) \; = \; \mathbf{e}_1' \, \hat{\beta}(x_0) \; = \; \ell(x_0)' \mathbf{Y}$$

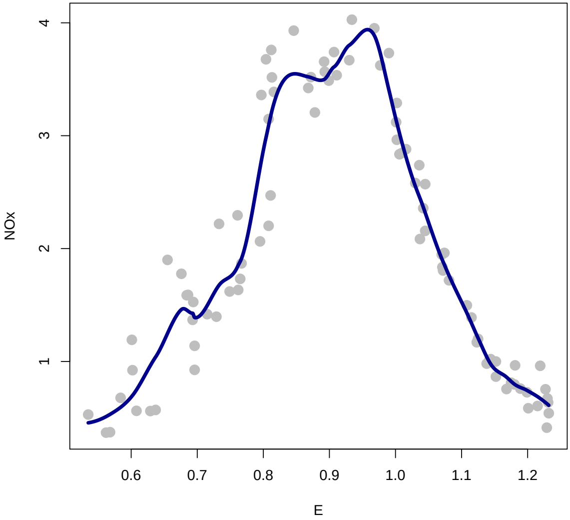
Varying bandwidths...

- This is what `loess` does in `R`
- Pay attention to the argument `family`
- What kernel does it use?
- Can you reproduce what `predict` does?

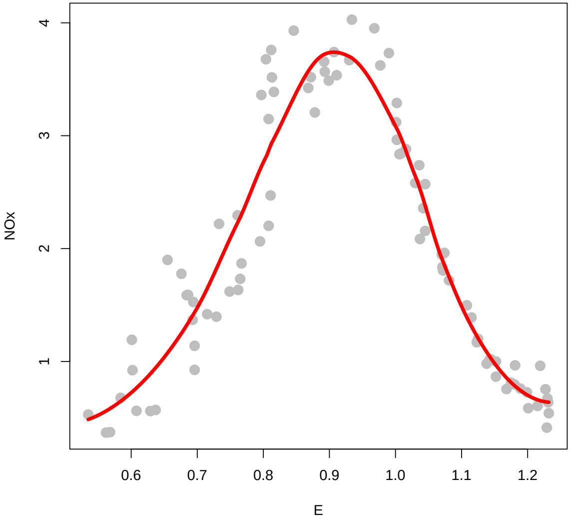
Varying bandwidths...

```
> plot(NOx ~ E, pch=19, col='gray', cex=1.5,
       data=ethanol)

> a <- loess(y~x, span=0.2, degree=2)

> xt <- seq(min(x), max(x), length=1000)

> yh <- predict(a, newdata=data.frame(x=xt))

> lines(xt, yh, lwd=4, col='darkblue')
```

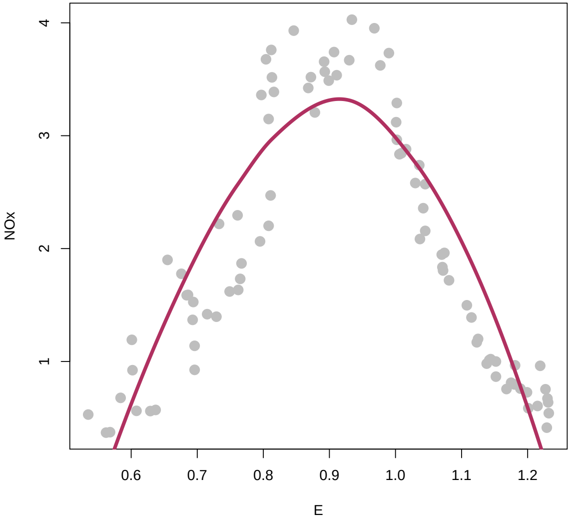
Degree=2, span = 0.2



Degree=2, span = 0.5



Degree=2, span = 1



Degree=2, **span = 2?**