Nonlinear regression

The general form of a nonlinear regression is:

$$y_k = h(\boldsymbol{x}_k, \boldsymbol{\beta}) + u_k,$$

where $h(\cdot, \cdot)$ is some unknown function of the regressors and parameters.

- $y = e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 x_2} e^{u}$ - can be linearized
- $y = \beta_0 + \beta_1 e^{\beta_2 x_1} + u$ - cannot be linearized
- $y = \beta_0 + \beta_1 x_1^{\gamma} + u$ - cannot be linearized

A popular special case of the non-linear regression is the single-index model

$$y_k = h(\boldsymbol{x}_k' \boldsymbol{\beta}) + u_k,$$

thus h is a function of a linear combination of the regressors.

Assumptions

- as before +
- $E(\boldsymbol{u}|\boldsymbol{X}) = \mathbf{0}$ is replaced with $E(u_i|h(\boldsymbol{x}_i, \boldsymbol{\beta})) = 0$: if u is uncorrelated with $\boldsymbol{x}$ it still may be correlated with some function of $\boldsymbol{x}$. In general $E(\boldsymbol{u}|\boldsymbol{X}) = \mathbf{0}$ is not needed.
- Identifiability of the model parameters: the model is identifiable if there is no a non-zero parameter $\boldsymbol{\beta}_0$, such that $h(\boldsymbol{x}_i, \boldsymbol{\beta}_0) = h(\boldsymbol{x}_i, \boldsymbol{\beta})$ for all $\boldsymbol{x}_i$.
  Note: in the linear regression it is sufficient to assume $rank(\boldsymbol{X}'\boldsymbol{X}) = J + 1$. Here it is not enough.

$$y = \frac{\beta_0 + \beta_1 x_1}{\beta_2 + \beta_3 x_2} + u. = \frac{2\beta_0 + 2\beta_1 x_1}{2\beta_2 + 2\beta_3 x_2} + u$$

$\Rightarrow \beta_0 - \beta_3$ cannot be identified uniquely

$\Rightarrow \beta_0 = 1.$ $\quad y = \frac{1 + \beta_1 x_1}{\beta_2 + \beta_3 x_2} + u$

$y = \beta_0 + \beta_1 \cdot e^{\beta_2 + \beta_3 x} + u. = \beta_0 + \beta_1 e^{\beta_2} \cdot e^{\beta_3 x} + u.$ $\quad \beta_4$

Estimation: the LS estimation can be used, but the asymptotic theory follows in a straightforward way from the quasi (!) maximum-likelihood estimation.

Assuming Gaussian residuals it holds:

$$\begin{aligned}
\mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}, u_k) &= \frac{1}{K}\sum_{k=1}^{K} \ln f(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}, u_k) \\
&= \frac{1}{K}\sum_{k=1}^{K} \ln\left\{\frac{1}{\sqrt{2\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(y_k - h(\boldsymbol{x}_k, \boldsymbol{\beta}))^2\right)\right\} \\
&= \frac{1}{K}\sum_{k=1}^{K} \left[-\ln\sqrt{2\sigma^2} - \frac{1}{2\sigma^2}(y_k - h(\boldsymbol{x}_k, \boldsymbol{\beta}))^2\right] \longrightarrow max
\end{aligned}$$

Thus the first order conditions for $\boldsymbol{\beta}$ are

$$\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}, u_k)}{\partial \boldsymbol{\beta}} = \sum_{k=1}^{K}(y_k - h(\boldsymbol{x}_k, \boldsymbol{\beta}))\frac{\partial h(\boldsymbol{x}_k, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \stackrel{!}{=} \mathbf{0}.$$

$\rightsquigarrow$ mostly a highly nonlinear system of equations solved numerically.

Consequences: since the resulting $\hat{\boldsymbol{\beta}}$ is a non-linear function of the residuals u_k

$E(\hat{\beta})$ – cannot be computed, since $\hat{\beta}$ has no explicit formula.

- ... the unbiasedness can not be proven in simple fashion;
- ... the variance of $\hat{\boldsymbol{\beta}}$ is not easy to derive;
- ... the exact distribution of $\hat{\boldsymbol{\beta}}$ is not Gaussian;
- ... all the inferences, like tests, are valid only asymptotically.

since $\hat{\beta}$ is not linear in u's any more.

but the ML estimators are consistent and efficient (they possess the smallest variance among all consistent and asymptotically normal estimators) $\Rightarrow$ still the best what we can get

Where all this comes from?

- Taylor expansion of $f(x)$ in neighborhood of x_0

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + ...$$

- Exact Taylor expansion of $f(x)$ in neighborhood of x_0 (mean-value theorem)

$$f(x) = f(x_0) + f'(x_+)(x - x_0),$$

where x_+ lies between x and x_0.

- $$\left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} = \left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}} + \left.\frac{\partial^2 \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}'}\right|_{\boldsymbol{\beta}_+} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$$

- $$\sqrt{K}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = -\left(\left.\frac{\partial^2 \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}'}\right|_{\boldsymbol{\beta}_+}\right)^{-1} \sqrt{K}\left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}}$$

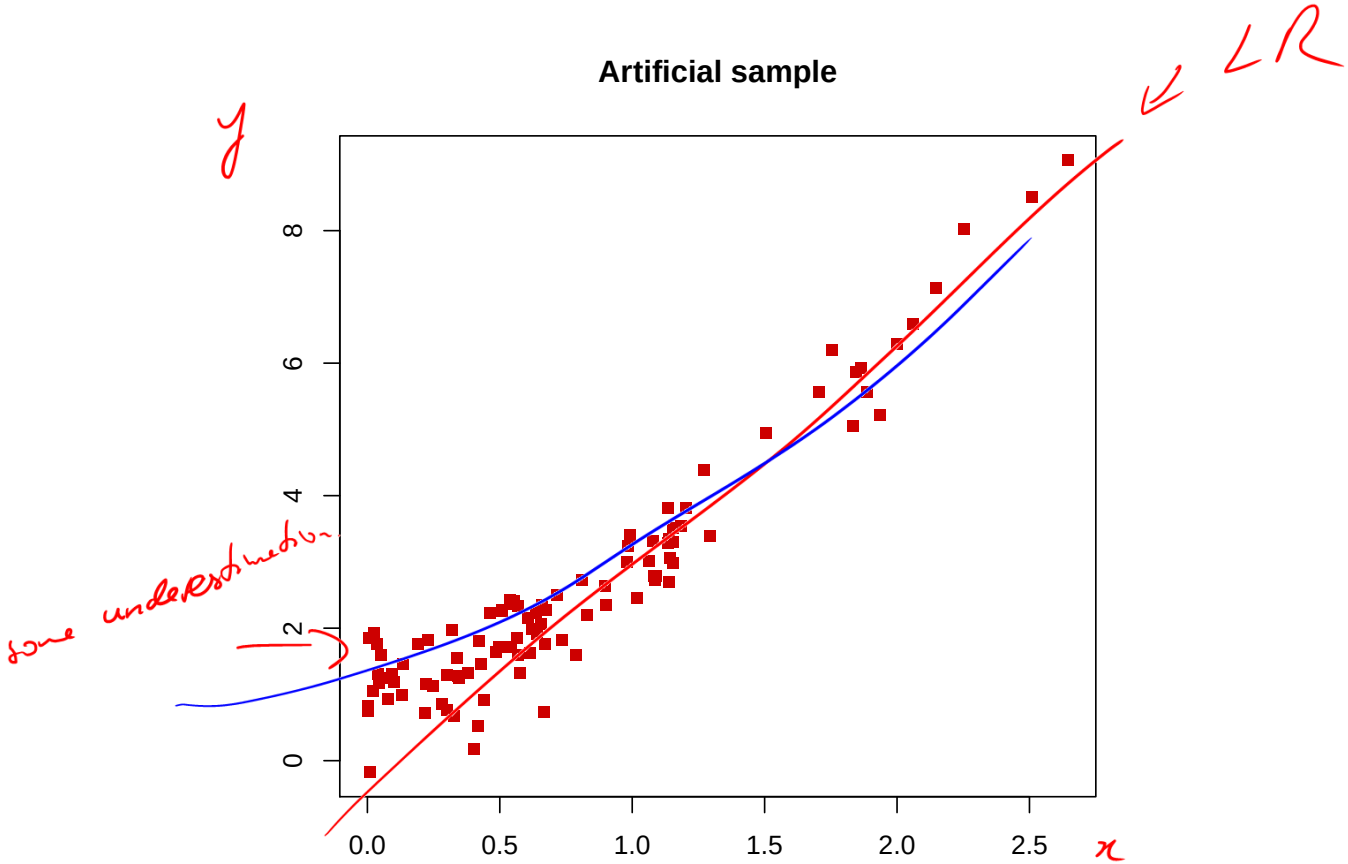
- $$\sqrt{K}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \overset{approx}{\sim} N(\mathbf{0}, \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{A}^{-1})$$

NLS

CLT

asymptotic variance

Example:



- Model 1 : $y = \beta_0 + \beta_1 x + u$
- Model 2 : $y = \beta_0 + \beta_1 x^{\beta_2} + u$

```
> z1 = lm(y~x);
> z2 = nls(y~ b0+b1*x^b2, start=list(b0=0, b1=1, b2=2))
> summary(z1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.44547    0.09694   4.595 1.29e-05 ***
x            2.78805    0.09787  28.488  < 2e-16 ***
---
Residual standard error: 0.604 on 98 degrees of freedom
Multiple R-squared:  0.8923,    Adjusted R-squared:  0.8912
F-statistic: 811.5 on 1 and 98 DF,  p-value: < 2.2e-16

> summary(z2)
Formula: y ~ b0 + b1 * x^b2

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0  1.08702    0.09345   11.63   <2e-16 ***
b1  1.77926    0.13041   13.64   <2e-16 ***
b2  1.56810    0.08382   18.71   <2e-16 ***
---

Residual standard error: 0.4678 on 97 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 2.905e-06
```

starting values

significant

R^2 high

rely on the asymptotic distribution (Slide 358).

all are significant

in LR.

$\Rightarrow$ nice to test if $H_0: \beta_2 = 1$ is rejected or not

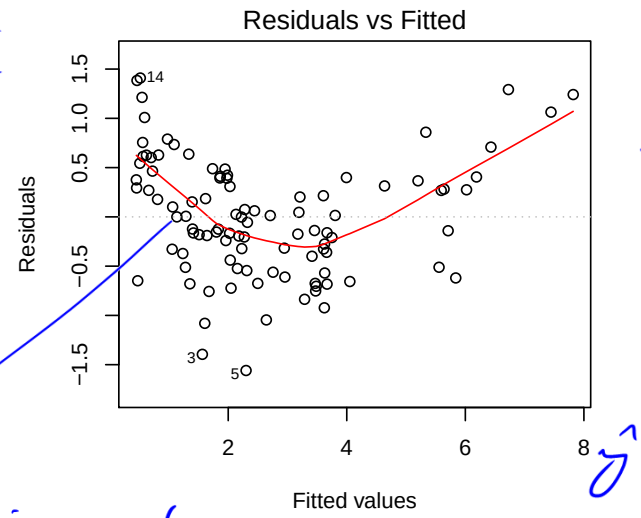
$\Rightarrow \hat{y} = 1{,}087 + 1{,}78 \cdot x^{1{,}56}$

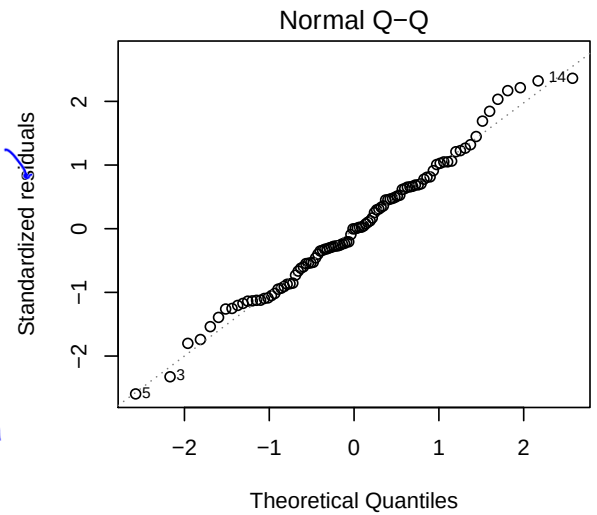
True model: $y = \underbrace{1 + 2x^{1.5}}_{\text{true value}} + u,\ u \sim N(0, 0.5^2).$ $n = 100$

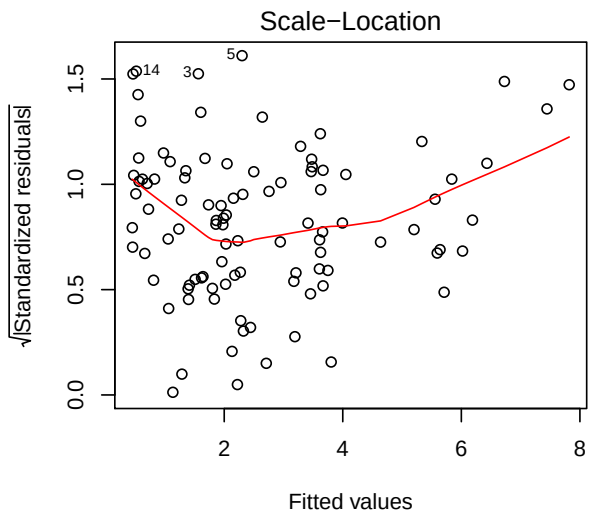
lm(y ~ x) linear.

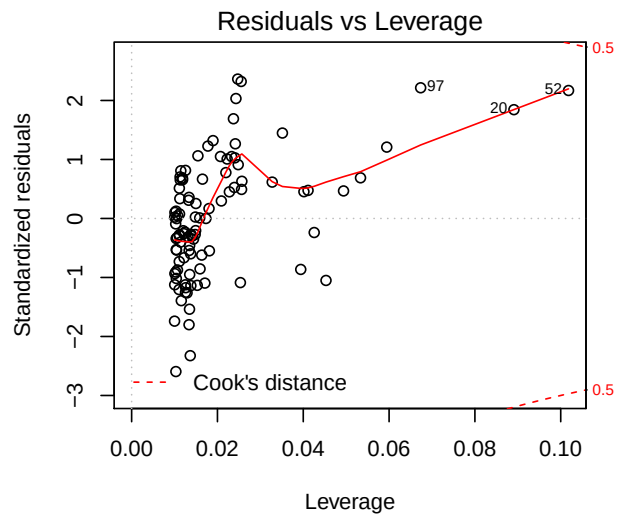
are random
=> no pattern allowed.

clear pattern
=> linear model is wrong!









Nonparametric regression

$$\{(X_i, Y_i)\},\ i = 1, \ldots, n; \quad \boldsymbol{X} \in \mathbb{R}^{J+1}, Y \in \mathbb{R}$$

- Engel curve: X = net-income, Y = expenditure

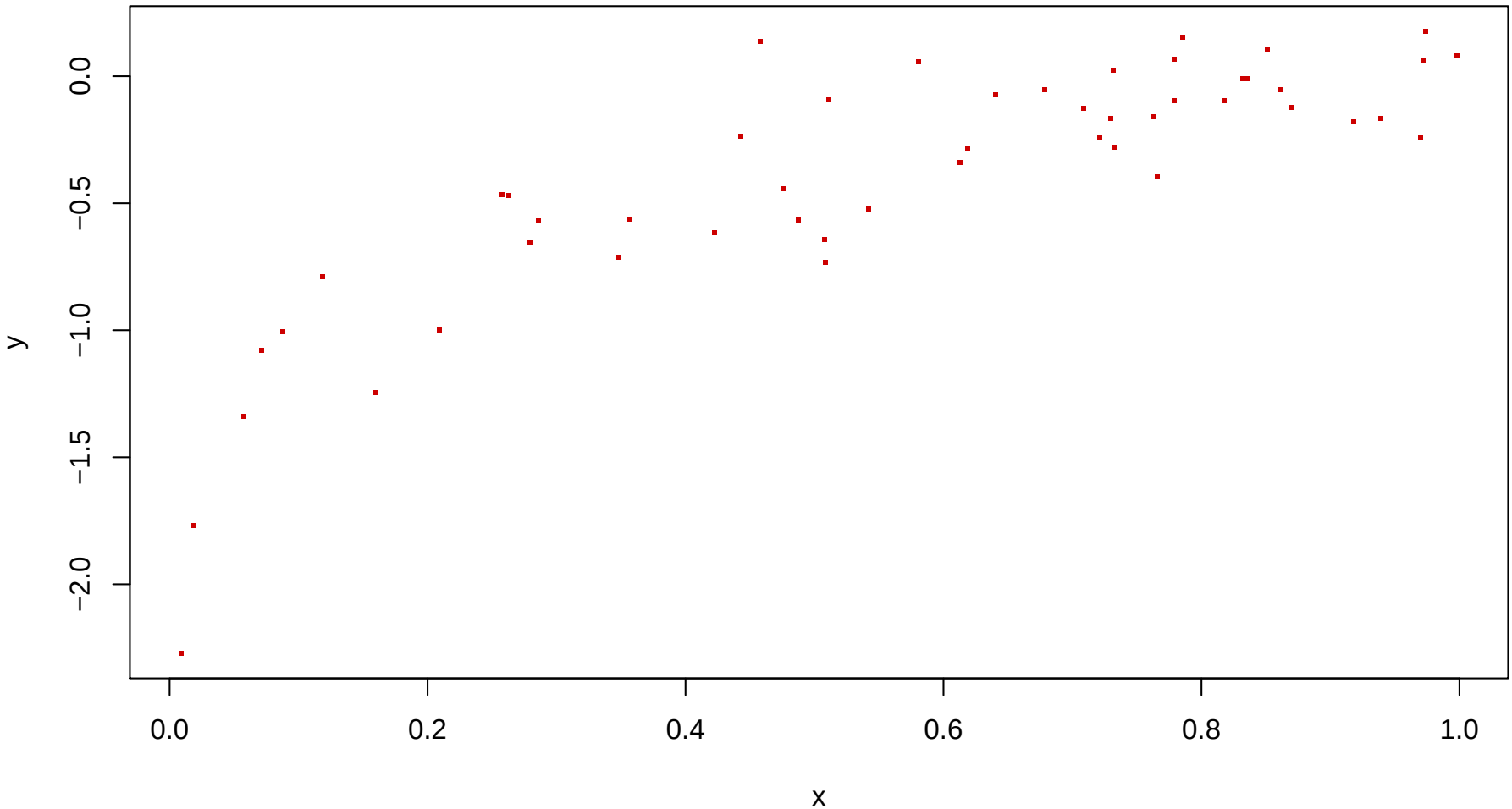
$$Y = m(X) + \varepsilon$$

- CHARN model: time series of the form

$$Y_t = m(Y_{t-1}) + \sigma(Y_{t-1})\xi_t$$

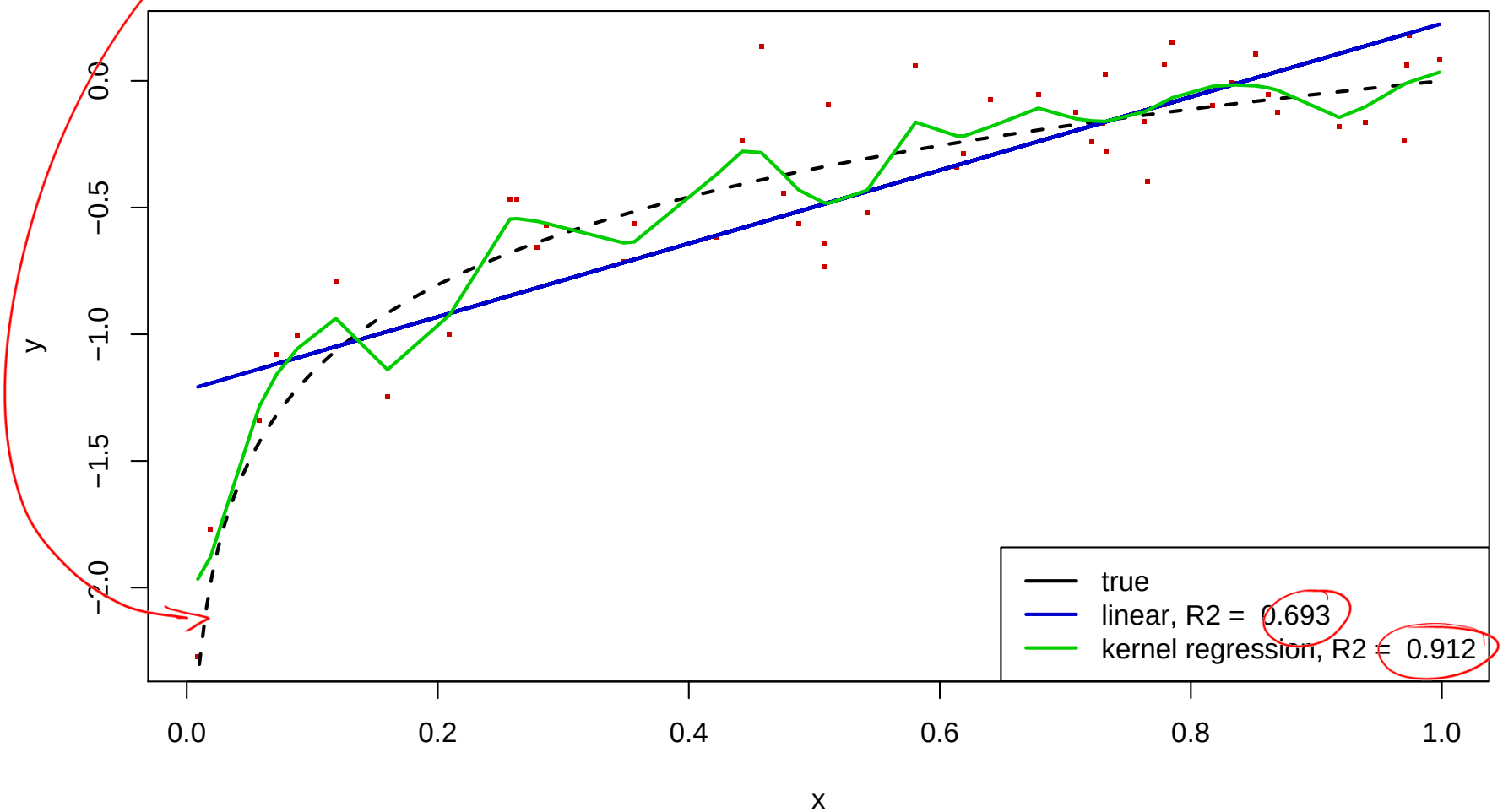
Note: Nonparametric regression is frequently used as a regularization technique in ML and is related to kNN

Let us have a sample of size $n = 50$ from an unknown model.

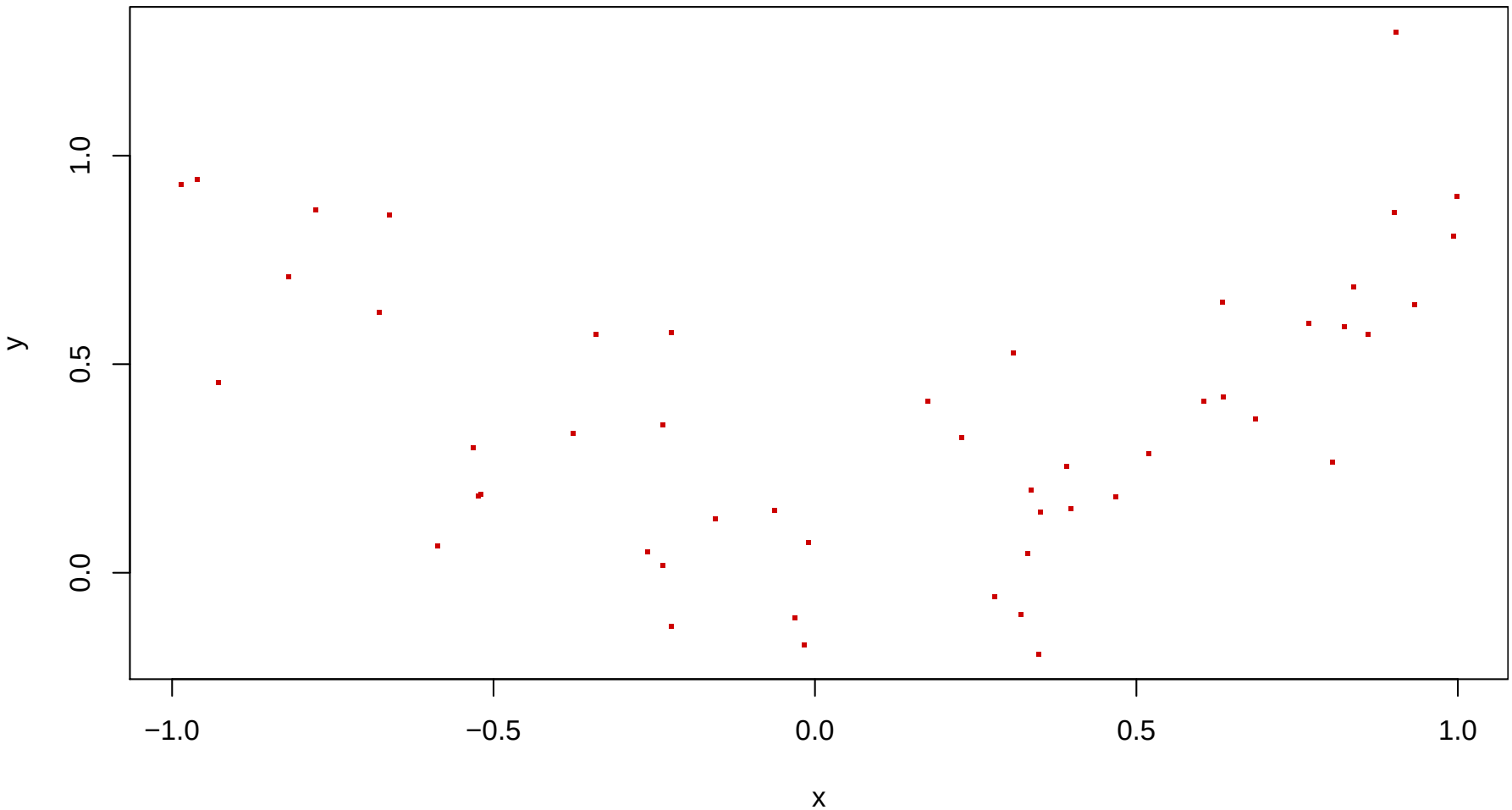


The data is simulated from the model

$$y_i = 0.5 \ln(x_i) + 0.2u_i, \qquad u_i \sim N(0,1).$$

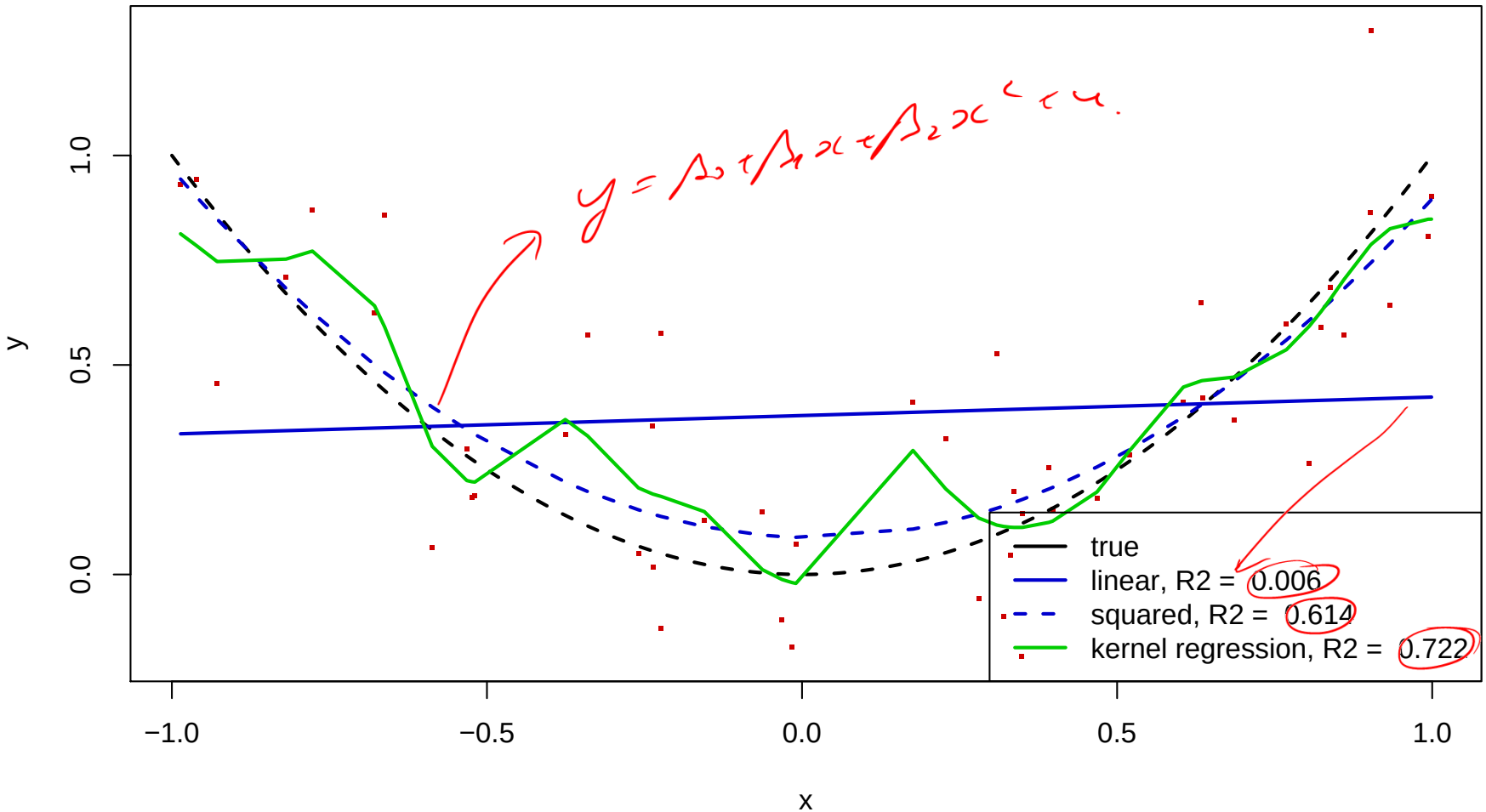


Another example.



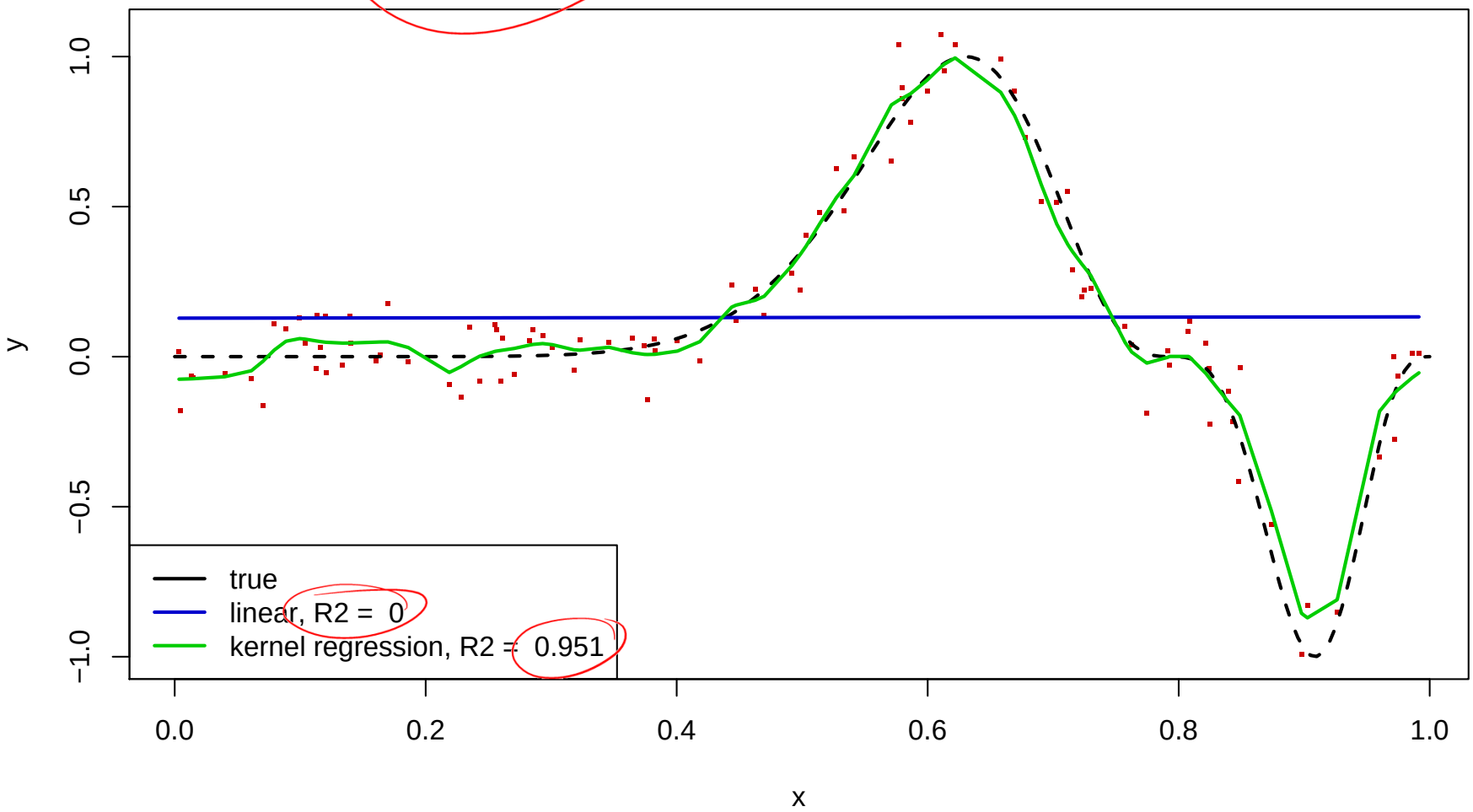
The data is simulated from the model

$$y_i = x_i^2 + 0.2u_i, \qquad u_i \sim N(0,1).$$



Hopeless example ...

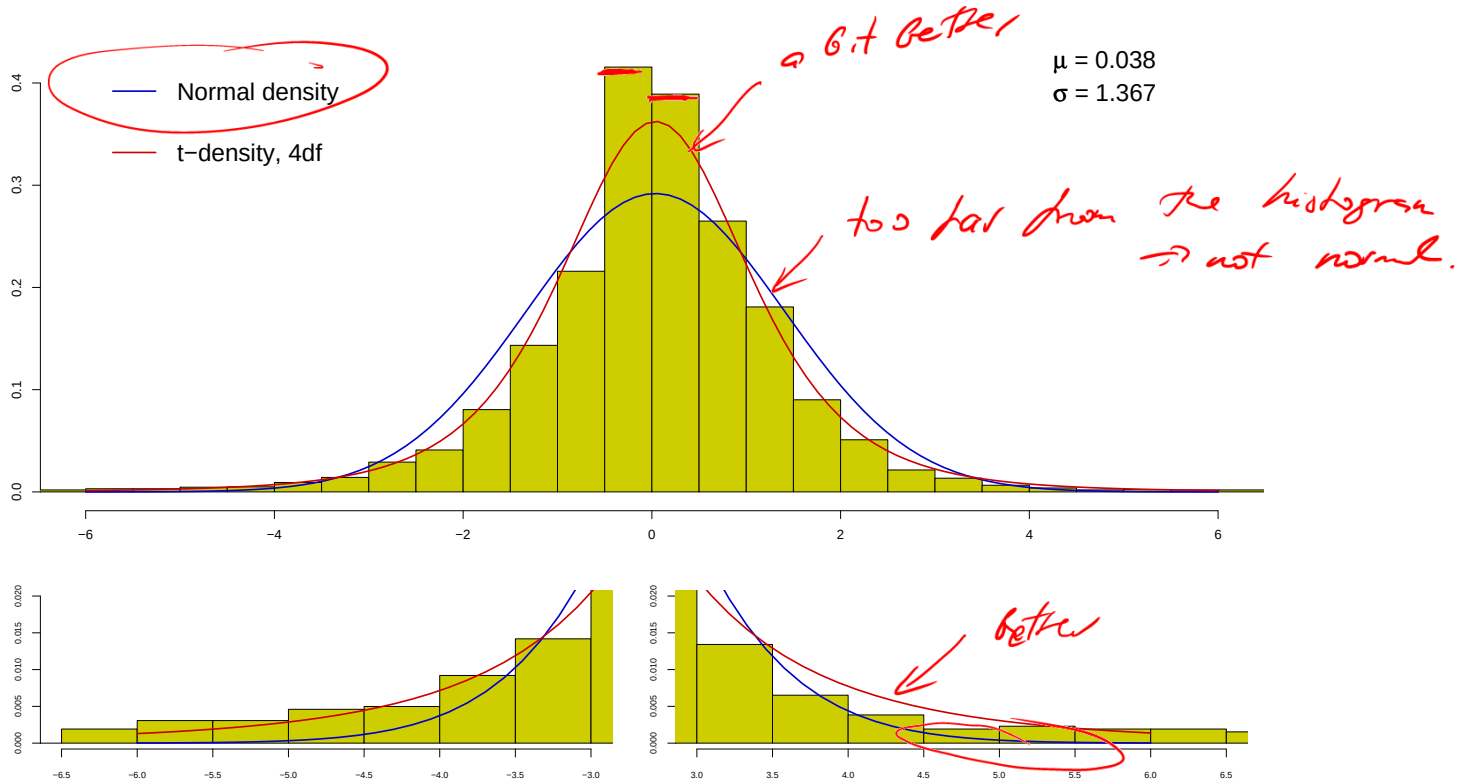
$$y_i = \{sin(2\pi x_i^3)\}^3 + 0.1u_i, \qquad u_i \sim N(0,1).$$



Kernel density estimator

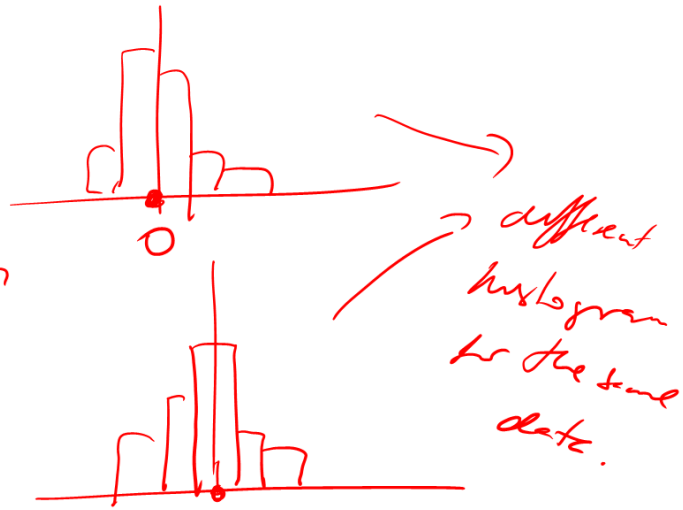
First: the procedure requires a non-parametric estimator of a density.

Here: DAX30 returns, 20 years of daily data, 5217 observations with normal and t-densities $\rightsquigarrow$ poor fit in the middle and in the tails

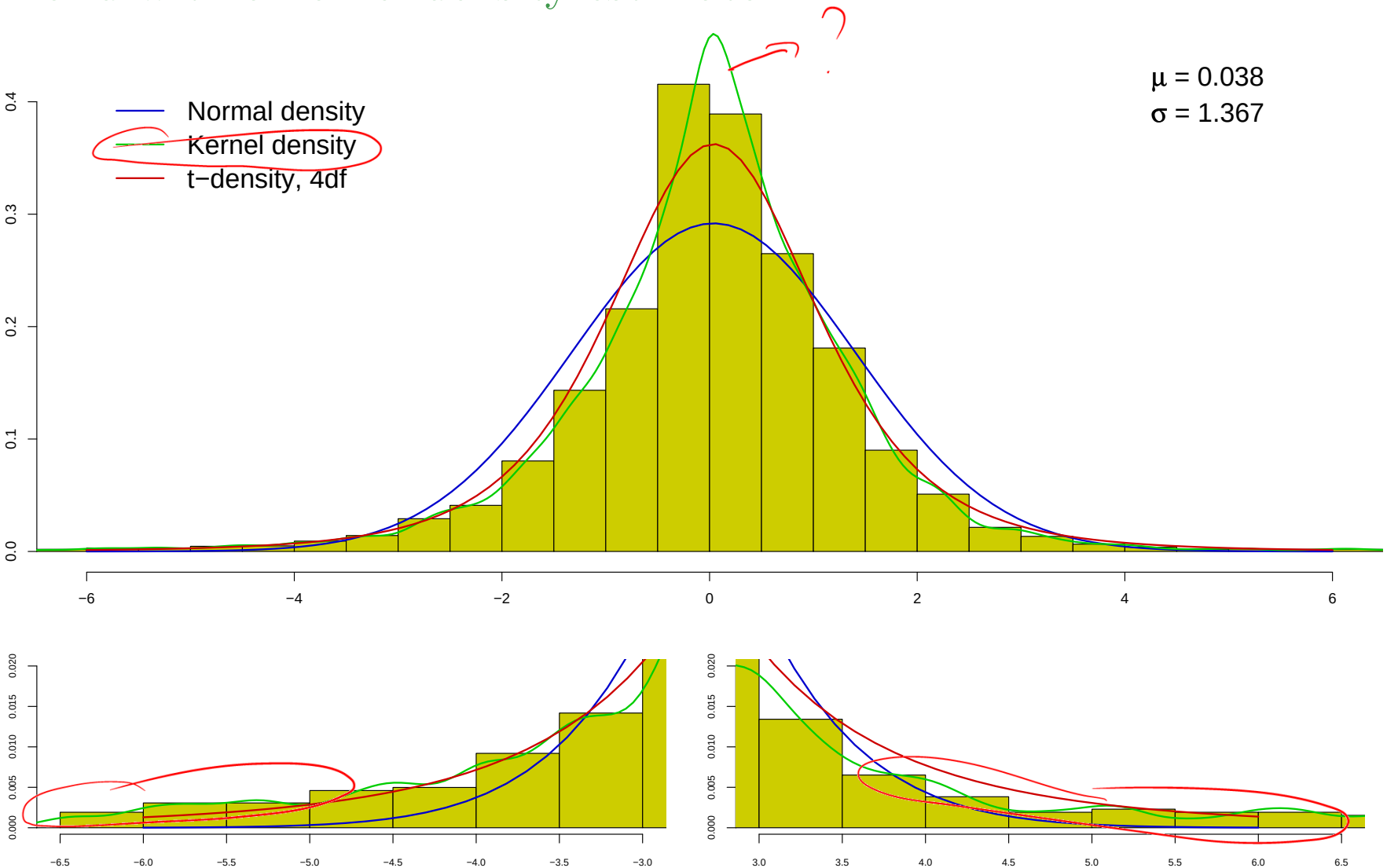


Drawbacks of the histogram

- constant over intervals, step function
- results depend strongly on origin
- binwidth choice
- (slow rate of convergence)



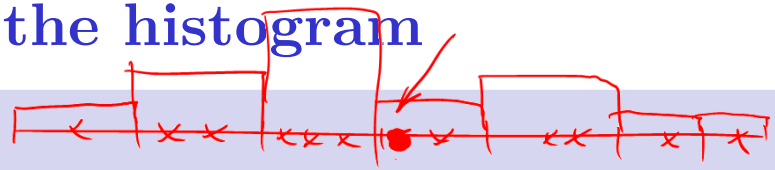
... and with a kernel density estimator



Kernel Density Estimation

KDE as a generalization of the histogram

Idea of the histogram:

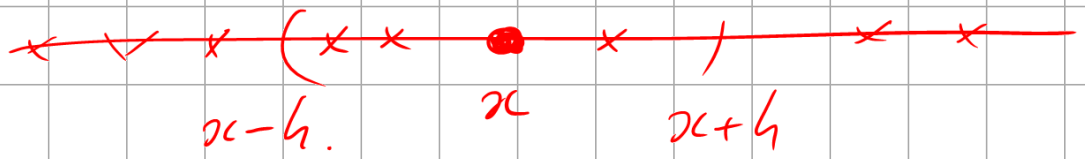


$$\frac{1}{n \cdot \text{interval length}} \#\{\text{obs. that fall into a small interval CONTAINING } x\}$$

Idea of the kernel density:

$$\frac{1}{n \cdot \text{interval length}} \#\{\text{obs. that fall into a small interval AROUND } x\}$$

$$\begin{aligned} \widehat{f}_h(x) &= \frac{1}{2hn} \sum_{i=1}^{n} I\left(\left|\frac{x - X_i}{h}\right| \leq 1\right) \\ &= \frac{1}{2hn} \#\{X_i \text{ in interval around } x\} \end{aligned}$$



$$I\left(x_i \in (x-h; x+h)\right) = \begin{cases} 1, & \text{if } x_i \in (x-h; x+h) \\ 0, & \text{else.} \end{cases}$$

$$\sum_{i=1}^{n} I\left(x_i \in \dots\right) \Rightarrow \text{number of observations falling into } (x-h; x+h)$$

$$x_i \in (x-h; x+h) \Rightarrow x-h < x_i < x+h$$

$$-h < x_i - x < h$$

$$-1 < \frac{x_i - x}{h} < 1$$

$$\Rightarrow \left| \frac{x - x_i}{h} \right| < 1$$

far away from x ⇒ small contribution

with I all three obstructions have the same impact/contribution

$$\sum I(\dots) = 0+0+0+0+1+1+1+0 \dots$$

⇒ Idea: replace I by a better function

kernel density estimate (KDE)

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

with kernel function $K(u)$

Required properties of kernels

- $K(\bullet)$ is a density function:

$$\int_{-\infty}^{\infty} K(u)du = 1 \quad \text{and} \quad K(u) \geq 0$$

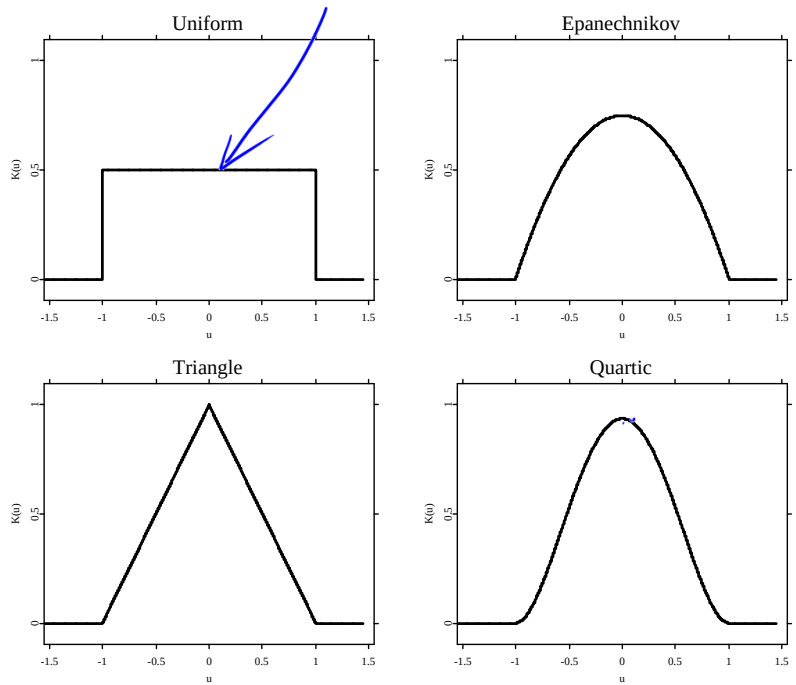
- $K(\bullet)$ is symmetric:

$$\int_{-\infty}^{\infty} uK(u)du = 0$$

Different Kernel Functions

indicator function I.

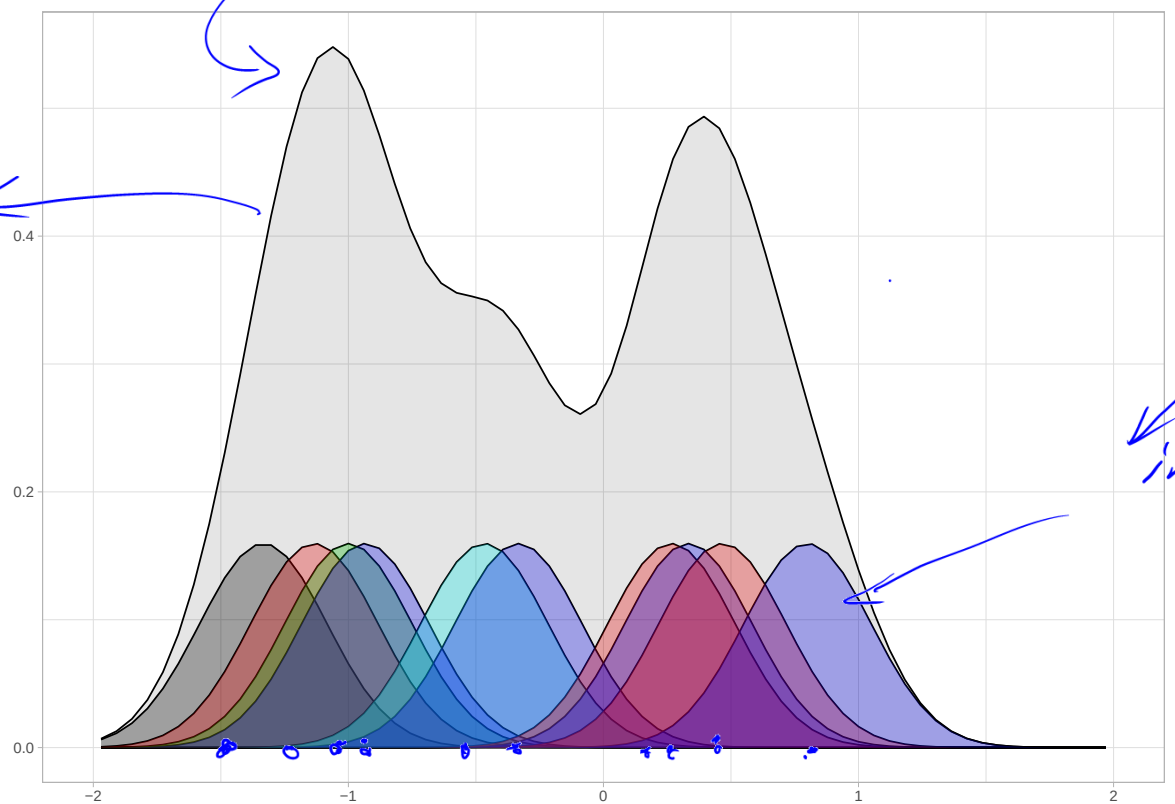
Kernel	$K(u)$
Uniform	$\frac{1}{2} I(\lvert u \rvert \leq 1)$
Triangle	$(1 - \lvert u \rvert) I(\lvert u \rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1 - u^2) I(\lvert u \rvert \leq 1)$
Quartic	$\frac{15}{16}(1 - u^2)^2 I(\lvert u \rvert \leq 1)$
Triweight	$\frac{35}{32}(1 - u^2)^3 I(\lvert u \rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} u^2)$
Cosinus	$\frac{\pi}{4} \cos(\frac{\pi}{2} u) I(\lvert u \rvert \leq 1)$



Example: Construction of the KDE

consider the KDE using a Gaussian kernel

$$\begin{aligned}\widehat{f}_h(x) &= \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \\ &= \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x - X_i)^2}{h^2}\right)\end{aligned}$$



Plotting kernel estimators in R

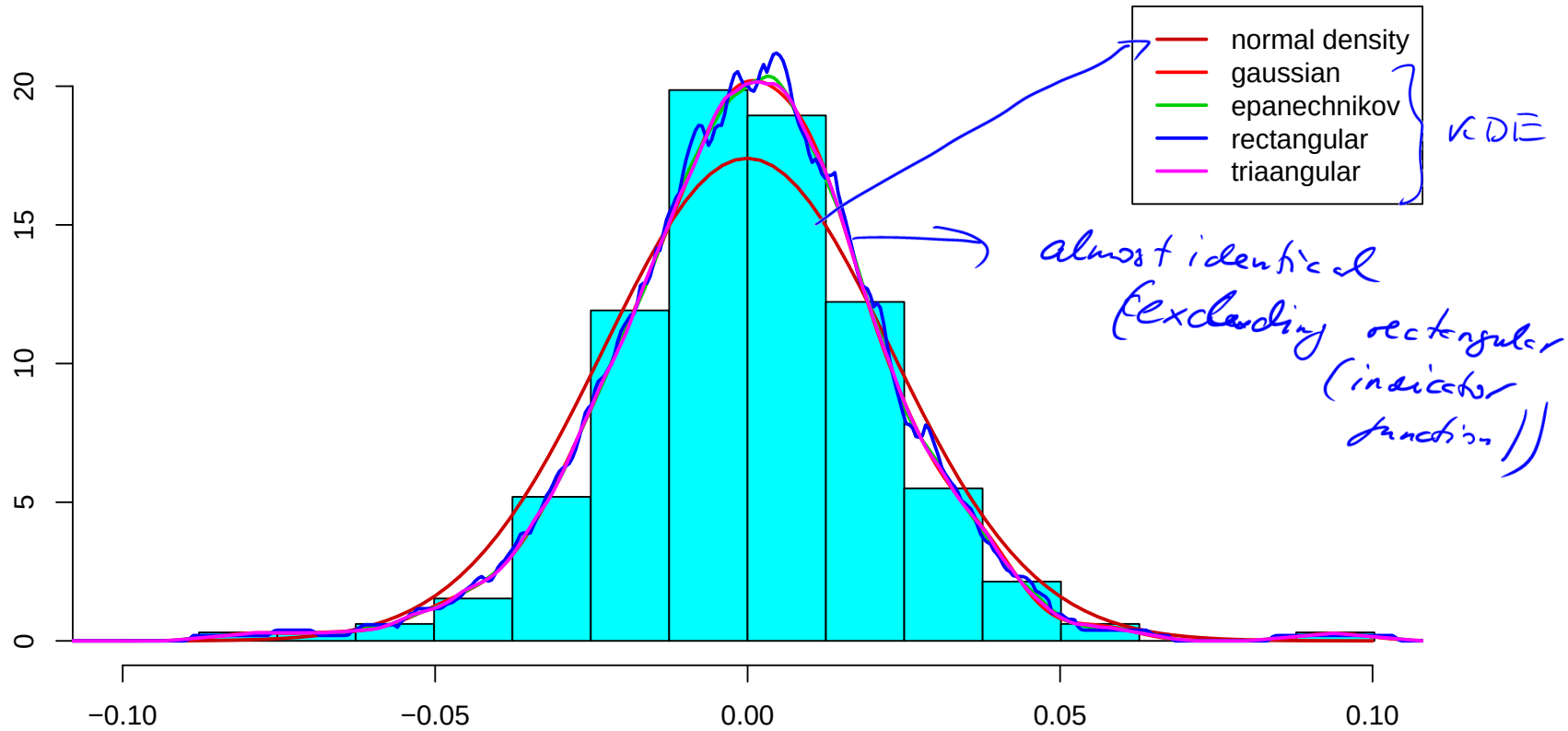
```
density(x, bw = "nrd0", adjust = 1,
        kernel = c("gaussian", "epanechnikov", "rectangular",
                   "triangular", "biweight", "cosine", "optcosine"),
        weights = NULL, window = kernel, width, give.Rkern = FALSE,
        n = 512, from, to, cut = 3, na.rm = FALSE, ...)
```

```
Value:
x: the 'n' coordinates of the points where the density is
          estimated.
y: the estimated density values.  These will be non-negative,
          but can be zero.
bw: the bandwidth used.
```

```
> x = rnorm(100)
> k = density(x)
> k
Data: x (100 obs.);     Bandwidth 'bw' = 0.3029
       x                   y
 Min.   :-3.4975   Min.   :0.000175
 1st Qu.:-1.7048   1st Qu.:0.023764
 Median : 0.0879   Median :0.061481
 Mean   : 0.0879   Mean   :0.139314
 3rd Qu.: 1.8806   3rd Qu.:0.248836
 Max.   : 3.6733   Max.   :0.443788
> k$x
  [1] -3.497498092 -3.483465207 -3.469432322 -3.455399437 -3.441366553
  [6] -3.427333668 -3.413300783 -3.399267898 -3.385235013 -3.371202128
  ...
> k$y
  [1] 0.0002050555 0.0002360509 0.0002706135 0.0003111481 0.0003565894
  [6] 0.0004070709 0.0004629191 0.0005277777 0.0005996858 0.0006789000
  ...
```

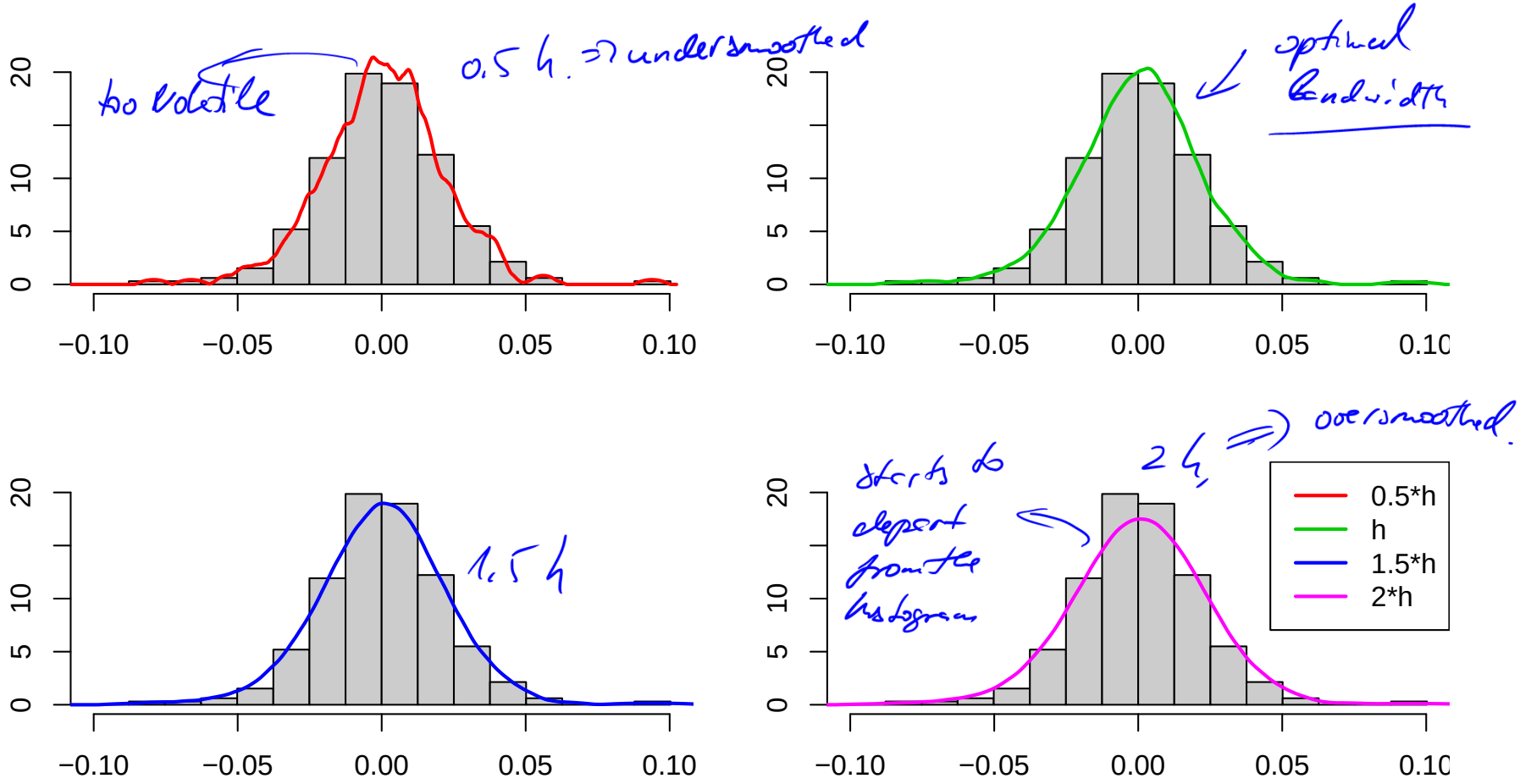
$\Rightarrow$ KDE depends on h and on K

Kernel estimator of the density of DJ30 returns with different kernels and "optimal" bandwidths.



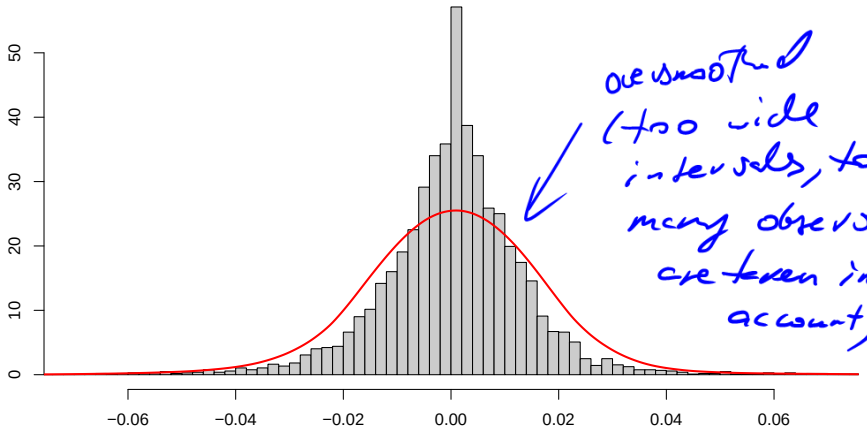
$\Rightarrow$ K has little impact on the KDE.

Epanechnikov kernel for the density of DJ30 returns with different adjusted "optimal" bandwidths.

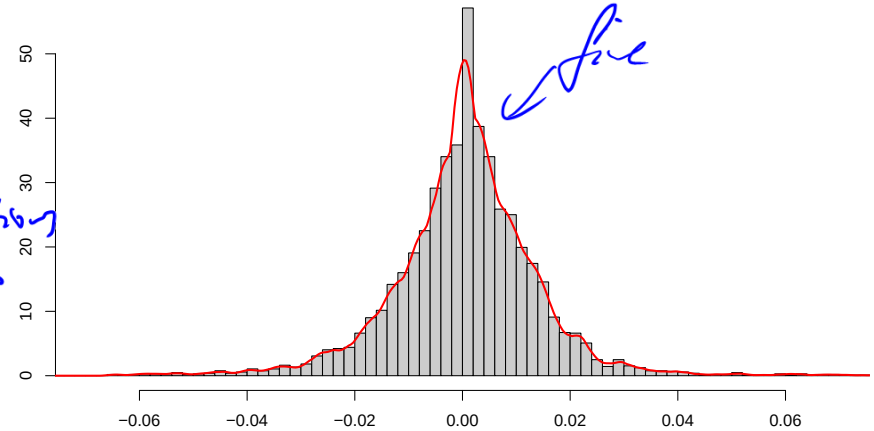


Epanechnikov kernel for the density of DAX30 returns with the bandwidths 0.01, 0.001, 0.0001.

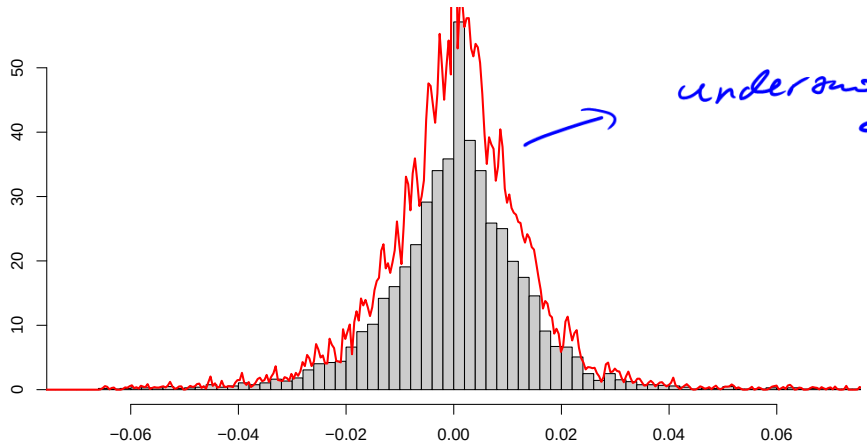
h.=



oversmoothed (too wide intervals, too many observations are taken into account)

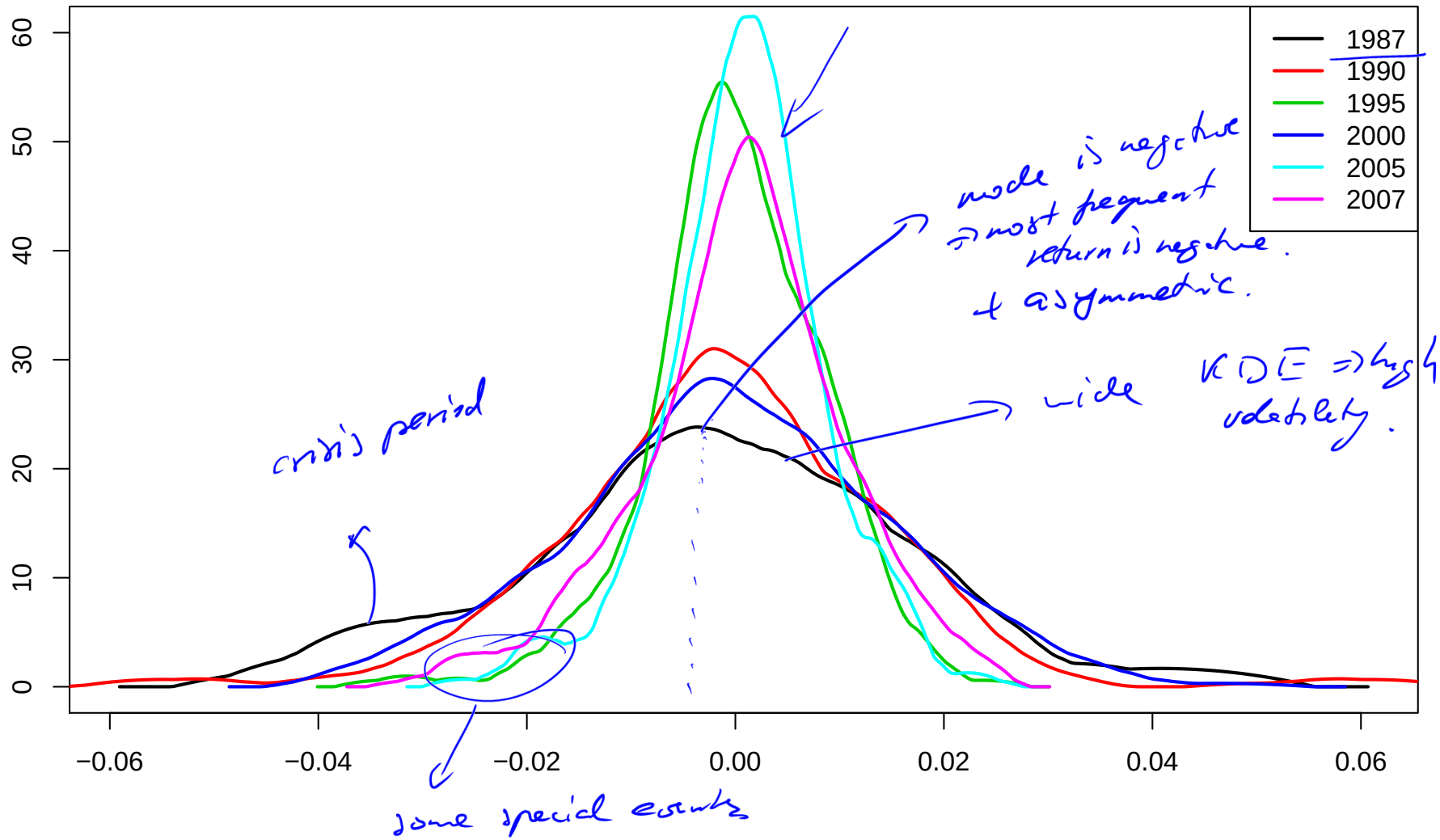


fine

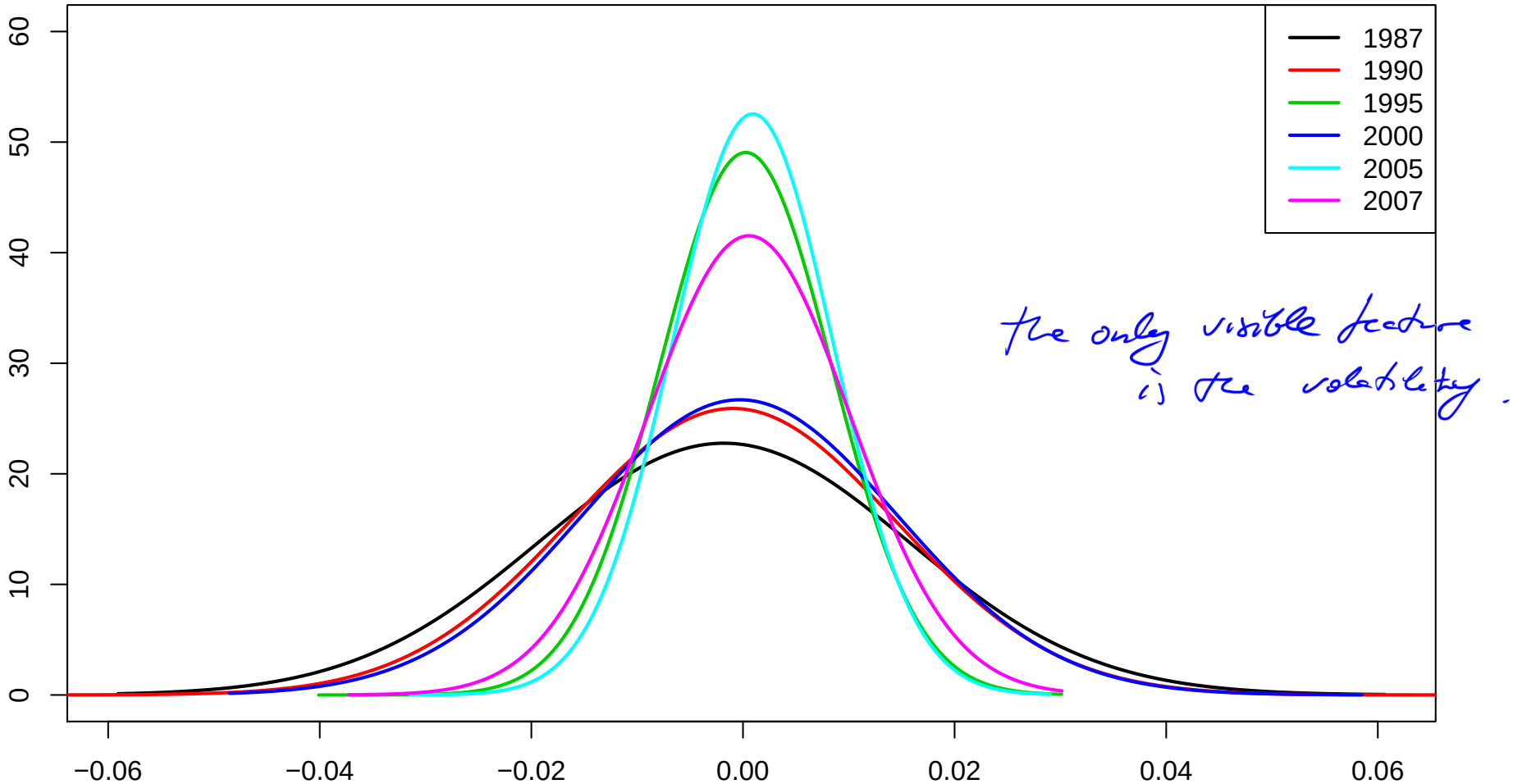


undersmoothed => the intervals around x are too small

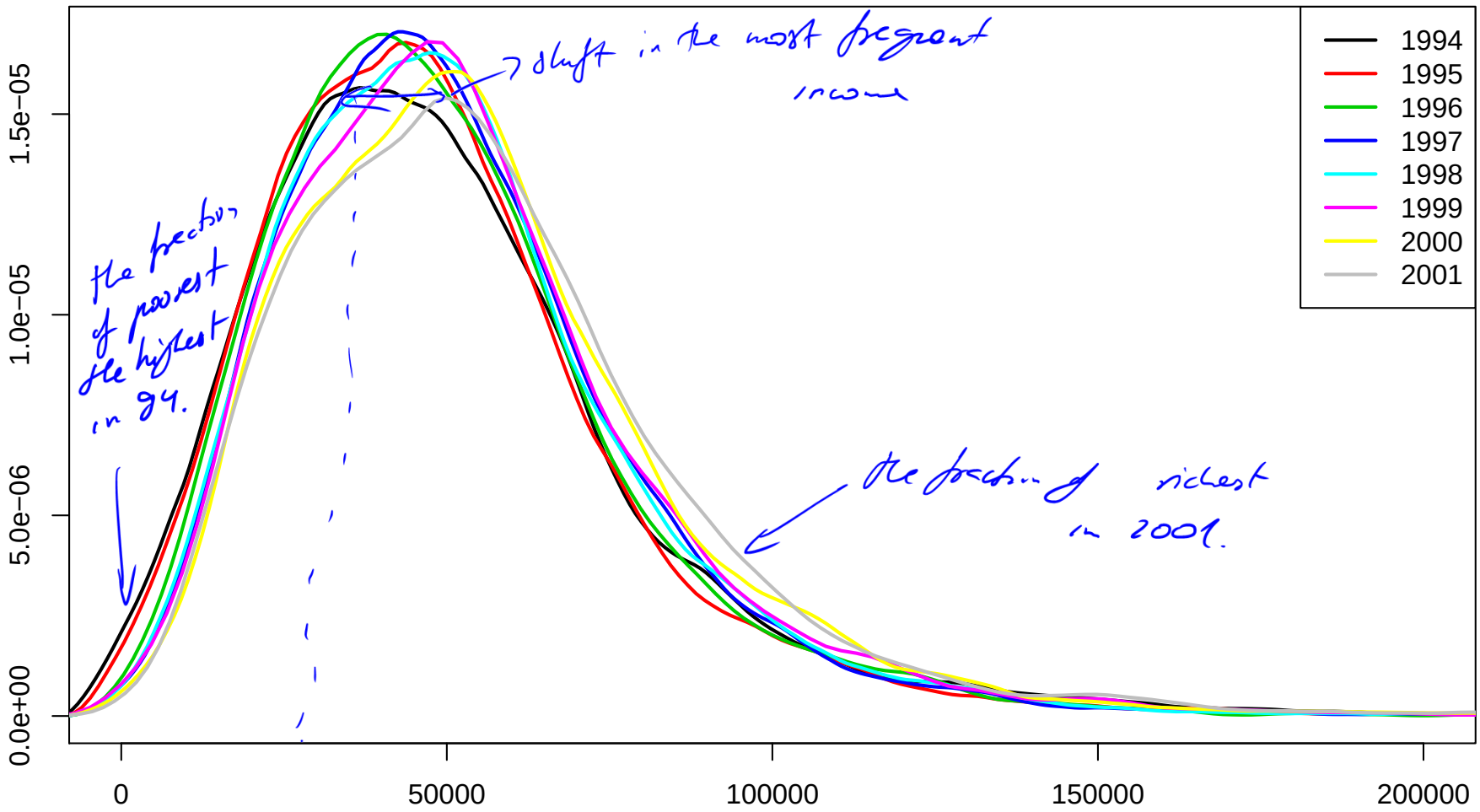
Example: time-variation of the distribution of DAX returns (Epanechnikov kernel)



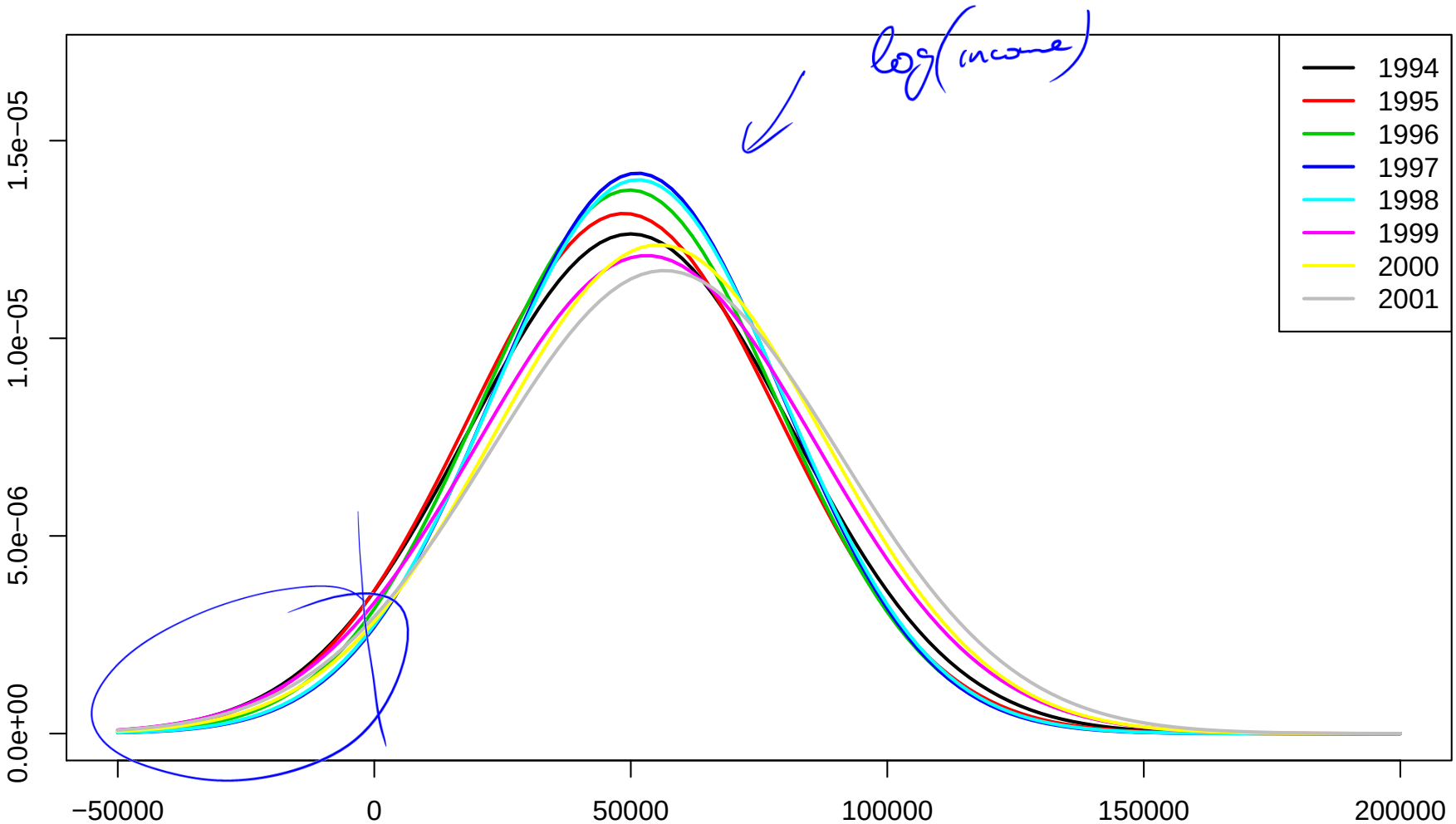
Example: time-variation of the distribution of DAX returns (normal density)



European Community Household Panel for the period 1994-2001. Data for Germany, net household income, ca. 48000 observations and Epanechnikov kernel.



European Community Household Panel for the period 1994-2001. Data for Germany, net household income, ca. 48000 observations and gaussian density .



(Asymptotic) statistical properties of KDE bias of the kernel density estimator

$$\begin{aligned} Bias\left\{\widehat{f}_h(x)\right\} &= E\left[\widehat{f}_h(x)\right] - f(x) \\ &= E\left[\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\right] - f(x) \\ &\approx \frac{h^2}{2} f''(x)\mu_2(K) \quad \text{for } h \to 0 \end{aligned}$$

where $\mu_2(K) = \int_{-\infty}^{\infty} u^2 K(u) du$.

$$\begin{aligned} Var\left[\widehat{f}_h(x)\right] &= Var\left[\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\right] \\ &\approx \frac{1}{nh}\|K\|_2^2 f(x) \quad \text{for } nh \to \infty \end{aligned}$$

where $\|K\|_2^2 = \int_{-\infty}^{\infty} \{K(u)\}^2 du$.

How to choose the bandwidth for the KDE?

find the bandwidth which minimizes the $MISE$

$$\begin{aligned} MISE\left\{\widehat{f}_h(x)\right\} &= E\left[\int_{-\infty}^{\infty}\{\widehat{f}_h(x) - f(x)\}^2 dx\right] \\ &= \int_{-\infty}^{\infty} MSE[\widehat{f}_h(x)]dx \\ &\approx \frac{1}{nh}\|K\|_2^2 + \frac{h^4}{4}\mu_2(K)^2\|f''\|_2^2 = AMISE\left\{\widehat{f}_h(x)\right\} \end{aligned}$$

$\rightarrow$ min w.r.t. h.

and thus

$$h_{opt} = \left(\frac{\|K\|_2^2}{\|f''\|_2^2\mu_2(K)^2 n}\right)^{1/5} \sim n^{-1/5}$$

$\Rightarrow$ Silverman's rule of thumb

h - ?

$\hat{f}(x) - f(x)$ must be small ← choose h to minimize it!

$$E\left(\int_{-\infty}^{+\infty} [\hat{f}(x) - f(x)]^2 \, dx\right).$$

must be small.

as in OLS to get rid of >0 or <0.

we need a single h for all x's ⇒ one bandwidth for the whole curve.

⇒ integrate over all x's

$\hat{f}$ depends on a random sample

⇒ E to get rid of randomness.

Summary for KDE

- The KDE does not depend on the starting points of the classes.
- The resulting estimator is a smooth and continuous function.
- The estimator heavily depends on the bandwidth.
- The estimator is robust to different choices of the kernel function.
- The estimator is biased in general.
- Decreasing bandwidth implies smaller bias, but larger variance.

Univariate nonparametric regression

model

$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

$m(\bullet)$ smooth regression function, ε_i i.i.d. error terms with $E\varepsilon_i = 0$

we aim to estimate the conditional expectation of Y given $X = x$

$$m(x) = E(Y|X = x) = \int y\, f(y|x)\, dy = \int y\, \frac{f(x,y)}{f_X(x)}\, dy$$

where $f(x, y)$ denotes the joint density of (X, Y) and $f_X(x)$ the marginal density of X

Nadaraya-Watson Estimator

yesterday KDE for a univariate density

idea: (X_i, Y_i) have a joint pdf, so we can estimate $m(\bullet)$ by a multivariate kernel estimator

kernel for X — kernel for Y

$$\widehat{f}_{h,\widetilde{h}}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right) \frac{1}{\widetilde{h}} K\left(\frac{y - Y_i}{\widetilde{h}}\right)$$

individual bandwidths

and therefore $\int y\, \widehat{f}_{h,\widetilde{h}}(x,y)dy = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)Y_i$

resulting estimator:

$\widehat{m}(x)$ is linear in y_i! $\Rightarrow \widehat{m}(x) = \sum \omega_i \cdot y_i$

$$\widehat{m}_h(x) = \frac{n^{-1}\sum_{i=1}^{n} K_h(x - X_i)Y_i}{n^{-1}\sum_{j=1}^{n} K_h(x - X_j)} = \frac{\widehat{r}_h(x)}{\widehat{f}_h(x)}$$

$\hat{y} = X\hat{\beta} = X(X'X)^{-1}X'y \Rightarrow \hat{y}_i = \sum \omega_i^* y_i \Rightarrow$ LR leads also to linear forecasts specified, by OLS and linearity

$$\underline{\hat{m}(x)} = \int y \, \frac{\hat{f}(x,y)}{\hat{f}_X(x)} \, dy = \frac{1}{\hat{f}_X(x)} \cdot \int y \cdot \underbrace{\frac{1}{n} \sum_{i=1} \frac{1}{h} K\left(\frac{x - x_i}{h}\right)}_{\text{independent any}} \cdot \frac{1}{\tilde{h}} K\left(\frac{y - y_i}{\tilde{h}}\right) dy$$

$$= \frac{1}{\hat{f}_X(x)} \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \cdot K\left(\frac{x - x_i}{h}\right) \cdot \int \frac{1}{\tilde{h}} y \cdot K\left(\frac{y - y_i}{\tilde{h}}\right) dy$$

$$= \underline{\qquad\qquad} \cdot \left[\underbrace{\int \frac{y - y_i}{\tilde{h}} K\left(\frac{y - y_i}{\tilde{h}}\right) dy}_{=0, \text{ because } K \text{ is symmetric.}} + y_i \underbrace{\int \frac{1}{\tilde{h}} K\left(\frac{y - y_i}{\tilde{h}}\right) dy}_{=1, \text{ because } K \text{ is a density and the integral is } 1.} \right]$$

$$= \frac{\frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - x_i}{h}\right) \cdot y_i}{\hat{f}_X(x)} = \frac{\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)}.$$

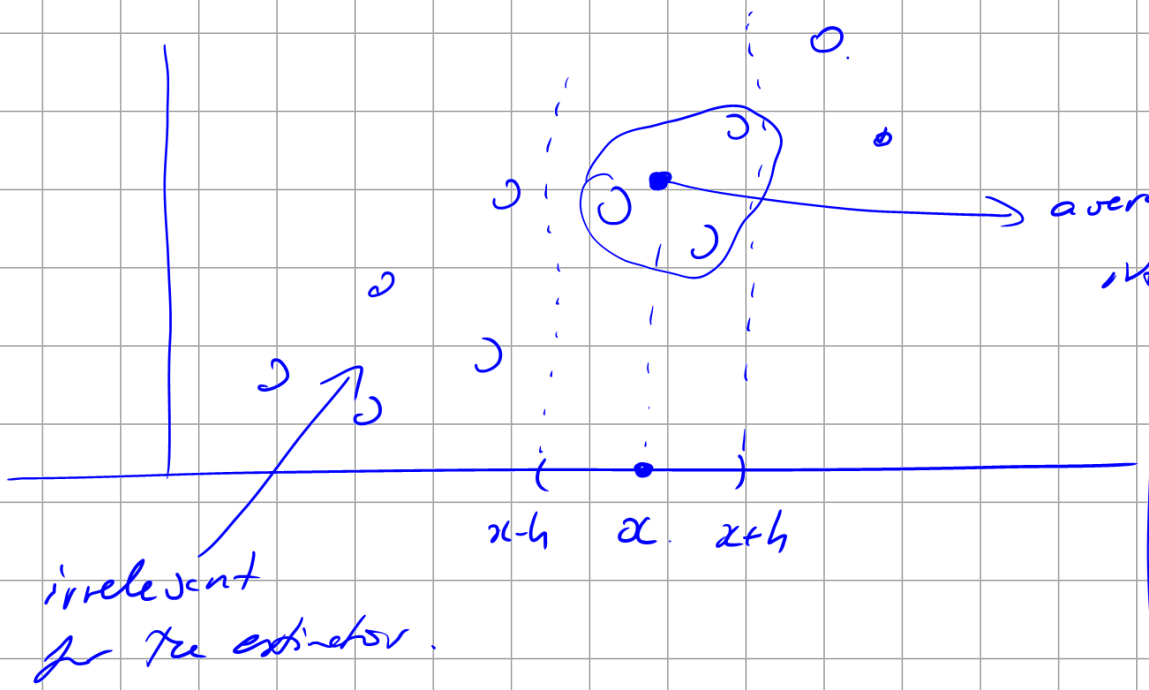
$\Rightarrow$ we get an estimator of m at x without any functional assumptions and using only KDE for densities.

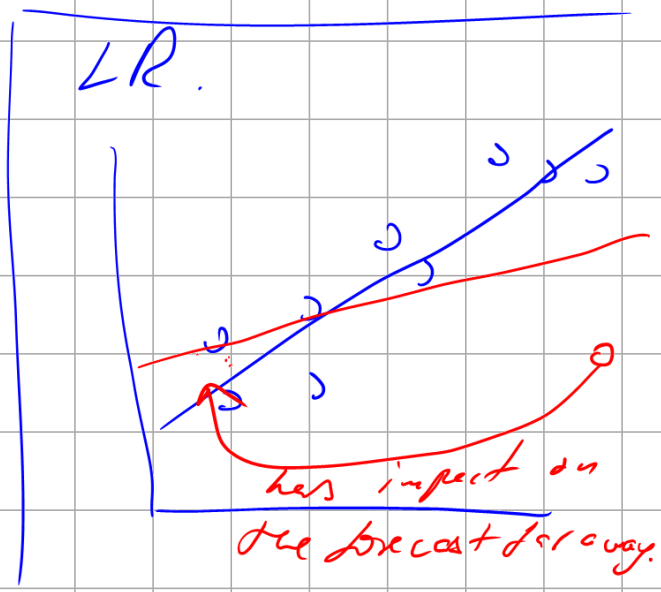
① no β's ② we have K (choice has little impact)

③ h (is important, but there are methods how to choose it)

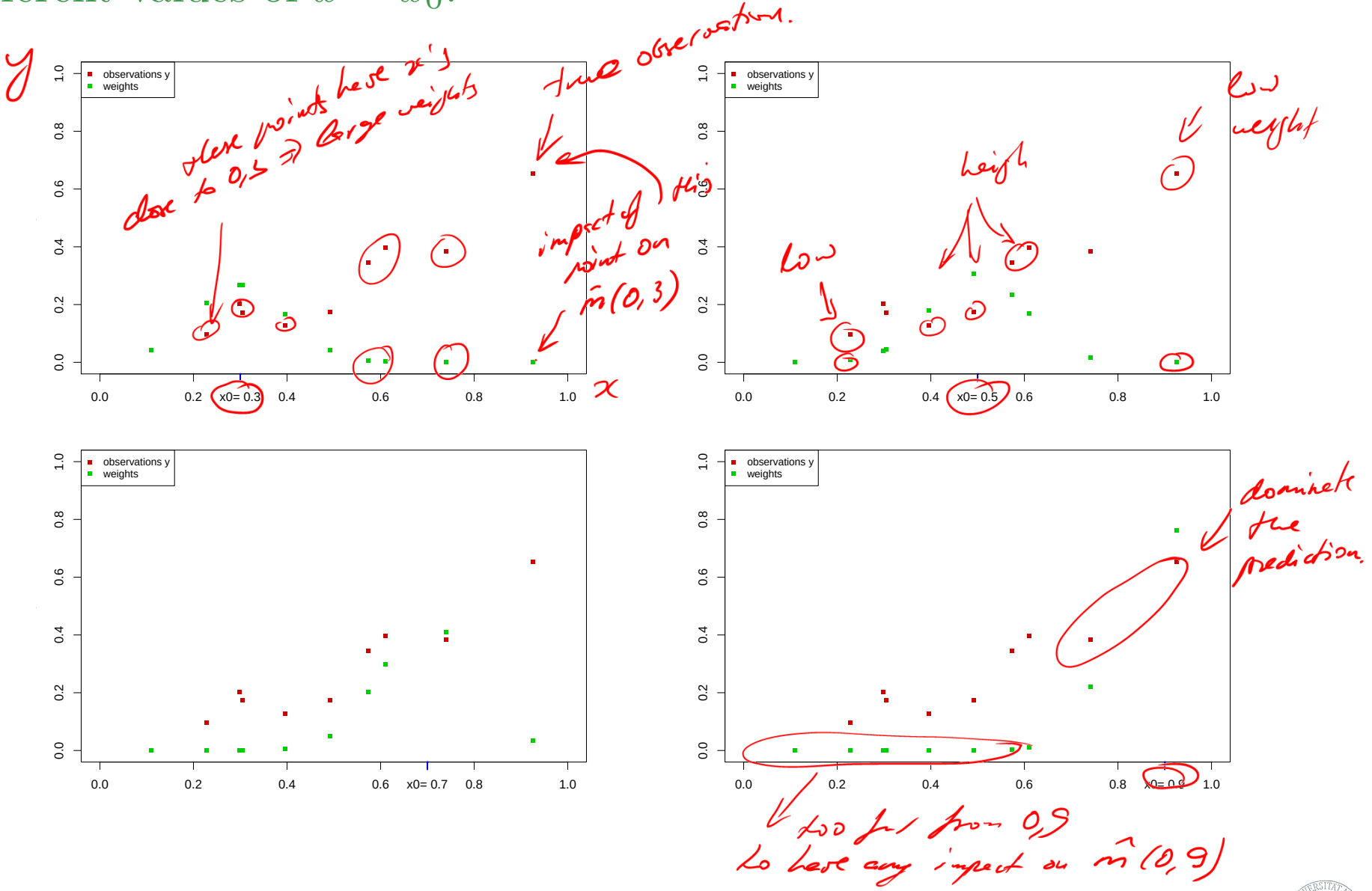
$\sum K\left(\frac{x - x_i}{h}\right) \cdot y_i$ = will weight only y's with x_i's close to x

$K\left(\frac{x - x_i}{h}\right)$ indicator $I\left(\left|\frac{x - x_i}{h}\right| \leq 1\right)$.





Observations and weights W_{hi} for $h = 0.1$, Gaussian kernel and different values of $x = x_0$.



Example: happiness

The happiness of nations (measured as average happiness of the citizens) depends of the average income per capita.

Data: 62 nations

$$\text{happiness} = m(\text{income}) + \varepsilon.$$

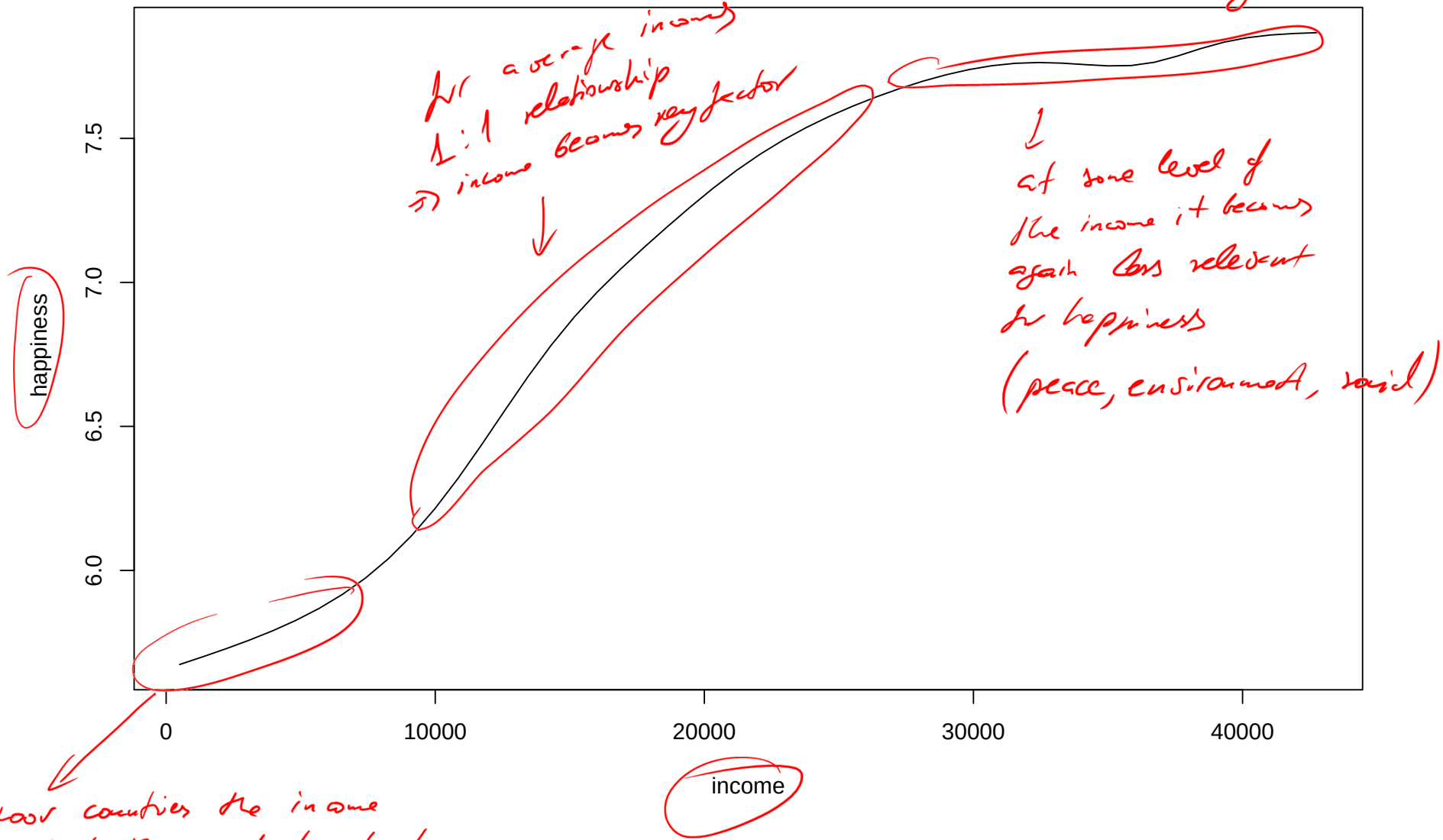
Nonparametric regression in R

```
library("np")
all = read.table("ch3_happiness.txt");
happiness = all[,1];
income = all[,2];

bw = npregbw(formula=happiness ~ income, lt="lc")
model = npreg(bws=bw);

npplot(bw, type="l");
```

$y = \beta_0 + \beta_1 \cdot g(x) + \varepsilon$ ⇒ not easy to get g, because of convexity and concavity.



for average income 1:1 relationship ⇒ income becomes very factor

↓ at some level of the income it becomes again less relevant for happiness (peace, environment, social)

poor countries the income is not the very factor for happiness (security, health)

Example: wage in Canada

Canadian cross-section wage data consisting of a random sample taken from the 1971 Canadian Census Public Use Tapes for male individuals having common education (Grade 13). There are n = 205 observations in total, and 2 variables, the logarithm of the individual's wage (logwage) and their age (age).

$$log(\text{wage}) = m(\text{age}) + \varepsilon$$

→ economically not clear.

$\log wage = \beta_0 + \beta_1 age + \varepsilon.$

```
data("cps71")
model.par = lm(logwage ~ age , data = cps71)
summary(model.par)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.021902   0.144705   89.99  < 2e-16 ***
age          0.012046   0.003554    3.39  0.00084 ***

Residual standard error: 0.6206 on 203 degrees of freedom
Multiple R-squared: 0.05357,    Adjusted R-squared: 0.04891
F-statistic: 11.49 on 1 and 203 DF,  p-value: 0.0008407
```

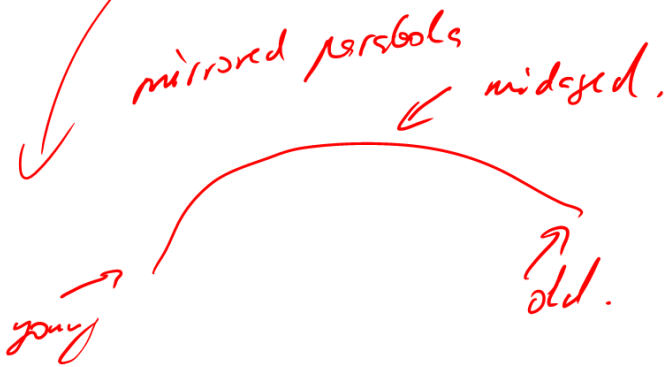
$\log\text{-wage} = \beta_0 + \beta_1 \cdot age + \beta_2 \cdot age^2 + \varepsilon$.

```
model.par2 = lm(logwage ~ age+I(age^2), data = cps71)
summary(model.par2
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.0419773  0.4559986  22.022  < 2e-16 ***
age          0.1731310  0.0238317   7.265 7.96e-12 ***
I(age^2)    -0.0019771  0.0002898  -6.822 1.02e-10 ***

Residual standard error: 0.5608 on 202 degrees of freedom
Multiple R-squared: 0.2308,     Adjusted R-squared: 0.2232
F-statistic:  30.3 on 2 and 202 DF,  p-value: 3.103e-12
```

< 0.05.



```
model.np = npreg(logwage ~ age, regtype = "lc", bwmethod = "cv.aic",
                 data = cps71)
summary(model.np)
```

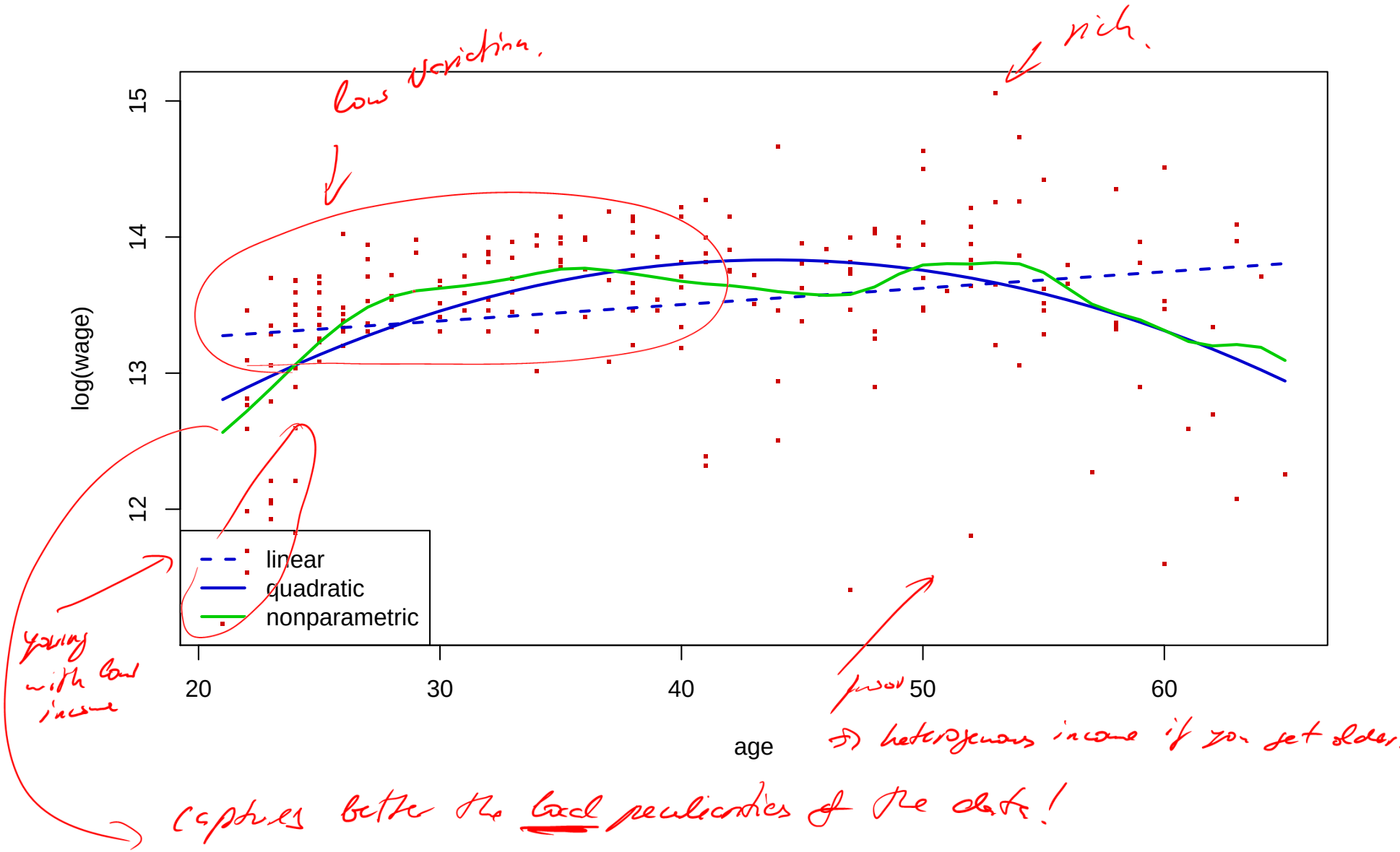
cross validation with AIC as a criterion. (alternative to Silverman's rule of thumb)

```
Regression Data: 205 training points, in 1 variable(s)
                     age
Bandwidth(s): 1.551218
```

$= h$

```
Kernel Regression Estimator: Local-Constant
Bandwidth Type: Fixed
Residual standard error: 0.2750934
R-squared: 0.3261299
```

$> 0{,}22 > 0{,}048$



Local regression.

classical regression is global: same line, same parameters for the whole data set

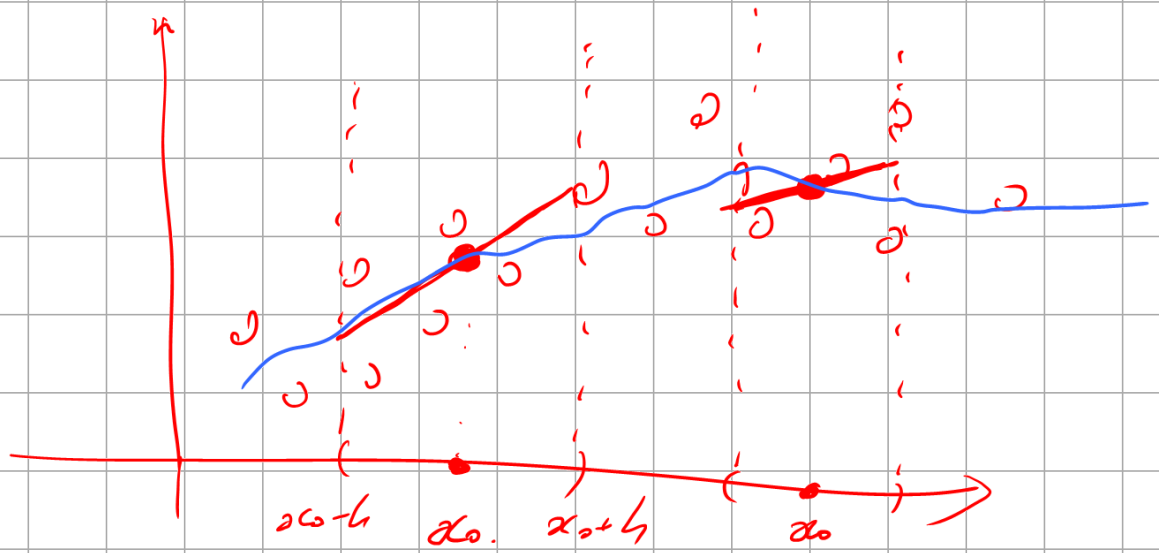
$$\sum (y_i - \beta_0 - \beta_1 x_i)^2 \to \min \text{ w.r.t. } \beta_0, \beta_1$$

we estimate regression at x_0.

$$\sum_{i=1}^{n} K\left(\frac{x_i - x_0}{h}\right) (y_i - \beta_0(x_0) - \beta_1(x_0) \cdot x_i)^2 \to \min \text{ w.r.t. } \beta_0(x_0), \beta_1(x_0)$$

$\to$ large ($\neq 0$) for x_i's close to x_0.

$\to$ small ($= 0$) for x_i's far away from x_0.



nonparametric : kernel

$\Rightarrow$ local modelling

$\Rightarrow$ smooth.

Lasso regression

Problems:

- accuracy: In K is much larger than J, then the variances are small and the inferences are precise. Low number of observations per parameter implies general high variability.
- interpretability: In large data sets there always irrelevant variables which make the economic interpretability difficult.
- sparsity: only a subset of the explanatory variables is relevant economically and statistically.

Solution: stepwise variable selection procedures based on statistical properties of the estimators or lasso regression

Idea: minimize the sum of squared residuals with constrains on the parameters. feature selection: some β's are set to zero

The objective function of the OLS procedure is replaced with

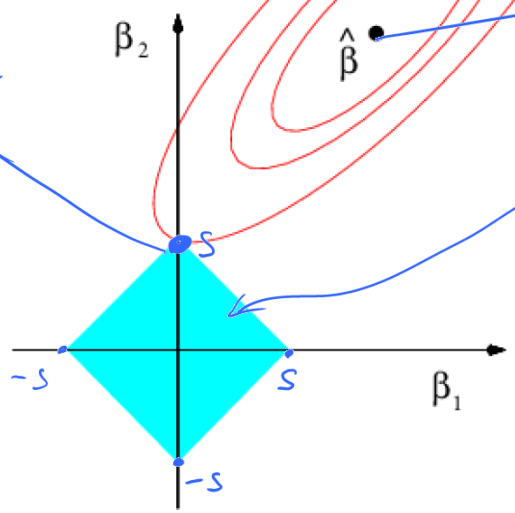
$$\sum_{k=1}^{K}(y_k - \beta_0 - \sum_{j=1}^{J}\beta_j x_{kj})^2 \quad + \lambda \sum_{j=1}^{J}|\beta_j| \longrightarrow min, \text{ w.r.t } \beta_j\text{'s}$$

without β_0, and with $|\cdot|$

OLS objective

penalty factor, which sets some β's to zero.

f cannot be minimized analytically



level lines of OLS objective

OLS estimator (bottom of 2dim. parabola)

solution of the constrained problem.

$\beta_1 = 0$, $\beta_2 \neq 0$

$\Rightarrow X_1$ is dropped

$\Rightarrow$ feature selection

$|\beta_1| + |\beta_2| \leq s$

$\beta_1 = 0 \Rightarrow -s \leq \beta_2 \leq s$

$\beta_2 = 0 \Rightarrow -s \leq \beta_1 \leq s$

Note:

- The problem is equivalent to the following problem, i.e. for each λ there exists s such that both problems lead to the same lasso-coefficients.

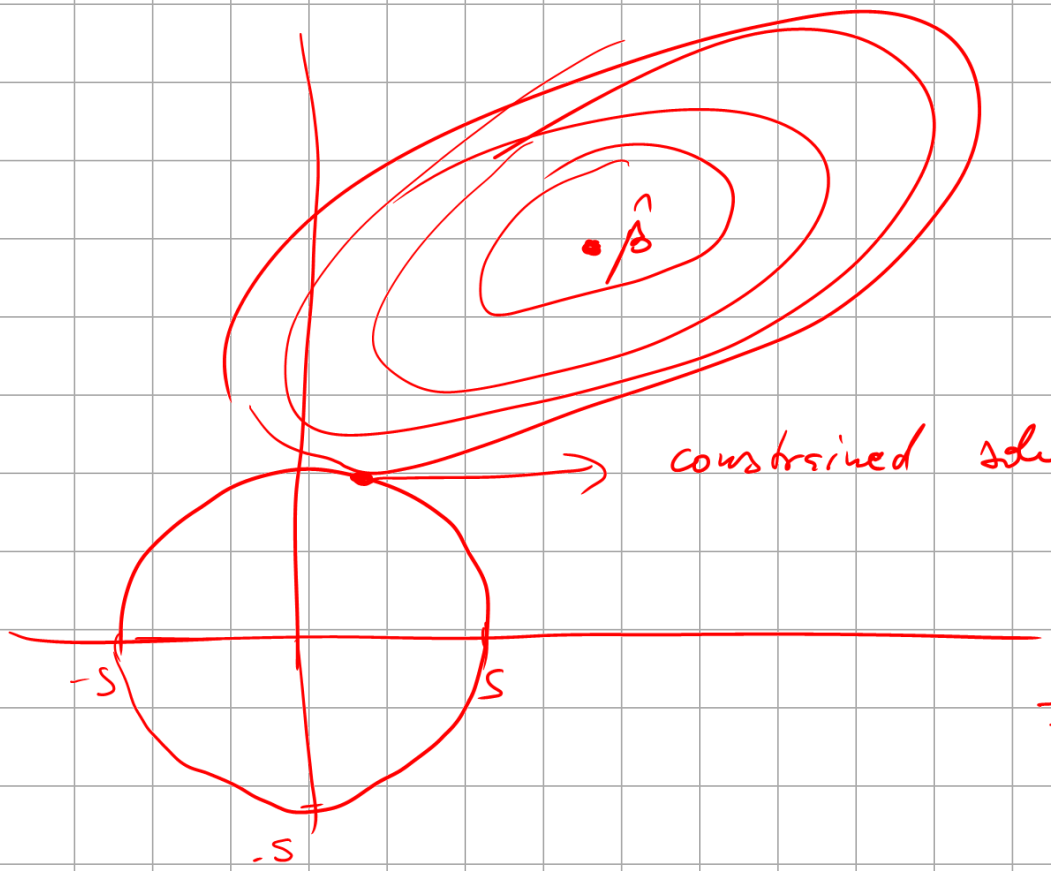
$$\sum_{k=1}^{K}(y_k - \beta_0 - \sum_{j=1}^{J}\beta_j x_{kj})^2 \longrightarrow min, \text{ w.r.t. } \beta_j\text{'s}$$

$$\text{s.t. } \sum_{j=1}^{J}|\beta_j| \leq s.$$

- Minimizing the objective is not trivial and there many specific numerical methods developed for this purpose.
- Selecting a good value for λ is crucial. The optimal value is chosen by cross-validation.
- typically one needs to standardize the inputs because we need $\sum |\beta_j|$.

Ridge regression $\Rightarrow$ in case of multicollinearity.

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 + \lambda \cdot \underbrace{\sum_{j=1}^{I} \beta_j^2}_{\text{square instead of } |\cdot|} \rightarrow \min, \quad \text{w.r.t. } \beta_j$$



$\beta_1^2 + \beta_2^2 \leq s \Rightarrow$ circle.

$\rightarrow$ constrained solution $\Rightarrow \beta_j$ are not set to zero, but are just shrunk to 0.

$\Rightarrow$ ridge is not a feature selection, but more a regularization.

Special case

dummies for every observation

$y_i = d_{1i} + d_{2i} + \dots + d_{Ki} + u_i \qquad d_{ki} = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$

Assume an individual constant for each observation:

$$\sum_{k=1}^{K} (y_k - \beta_k)^2 \Rightarrow \min \text{ w.r.t. } \beta_1 \dots \beta_K$$

with the OLS solution $\hat{\beta}_k = y_k$.

With lasso we obtain:

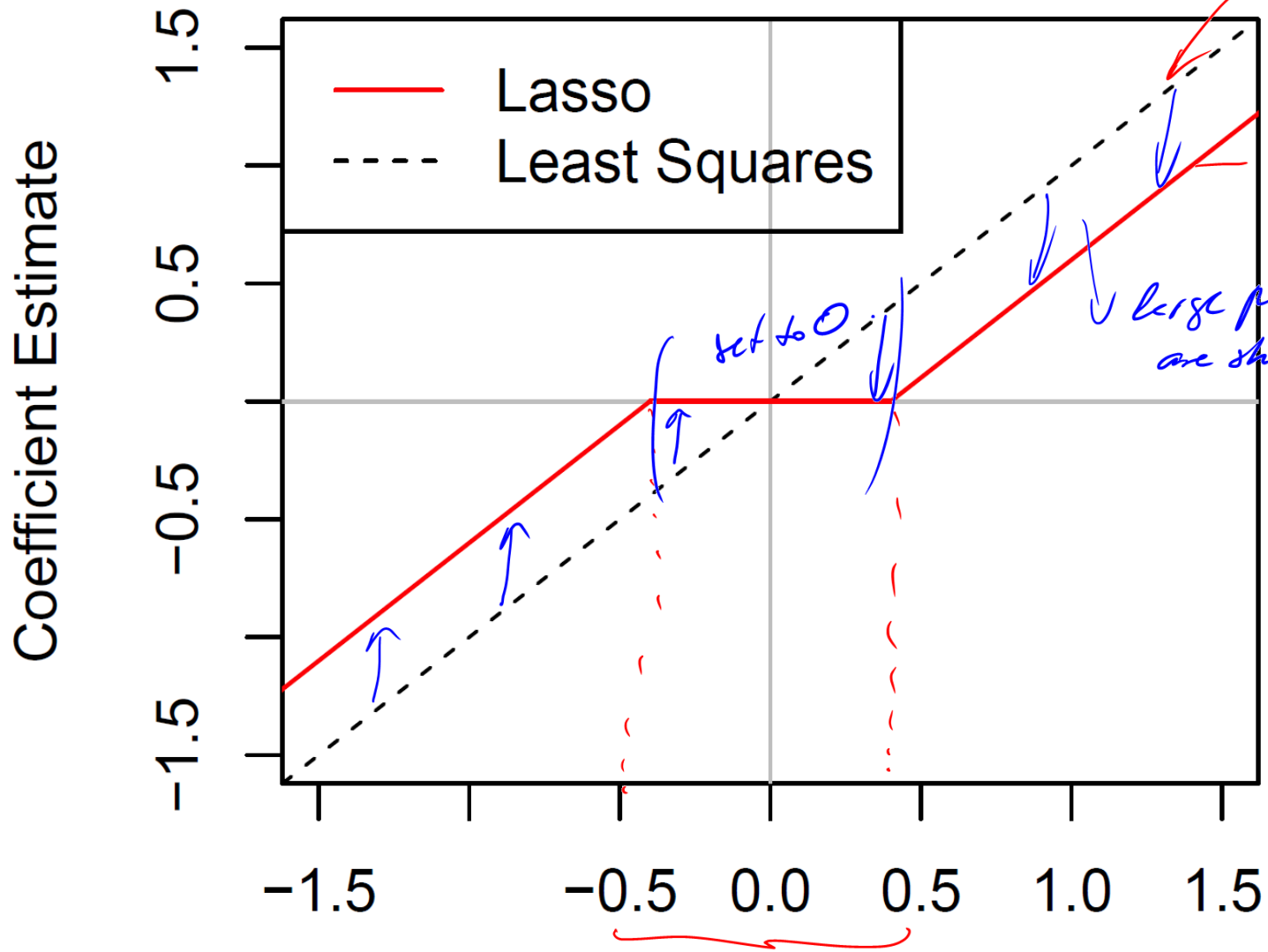
$$\sum_{k=1}^{K} (y_k - \beta_k)^2 + \lambda \sum_{k=1}^{K} |\beta_k| \longrightarrow min$$

with the solution

$$\hat{\beta}_k^{(lasso)} = \begin{cases} y_k - \lambda/2, & \text{if } y_k \geq \lambda/2 \\ y_k + \lambda/2, & \text{if } y_k \leq -\lambda/2 \\ 0, & \text{if } |y_k| \leq \lambda/2 \end{cases}$$

large observations

small observations



$\hat{\beta}_k = y_k$ OLS.

set to 0

large parameters are shrunk to 0.

small parameters.

⇒ ! not only small parameters are set to 0, but also large parameters are shrunk towards 0. ⇒ Biased OLS.

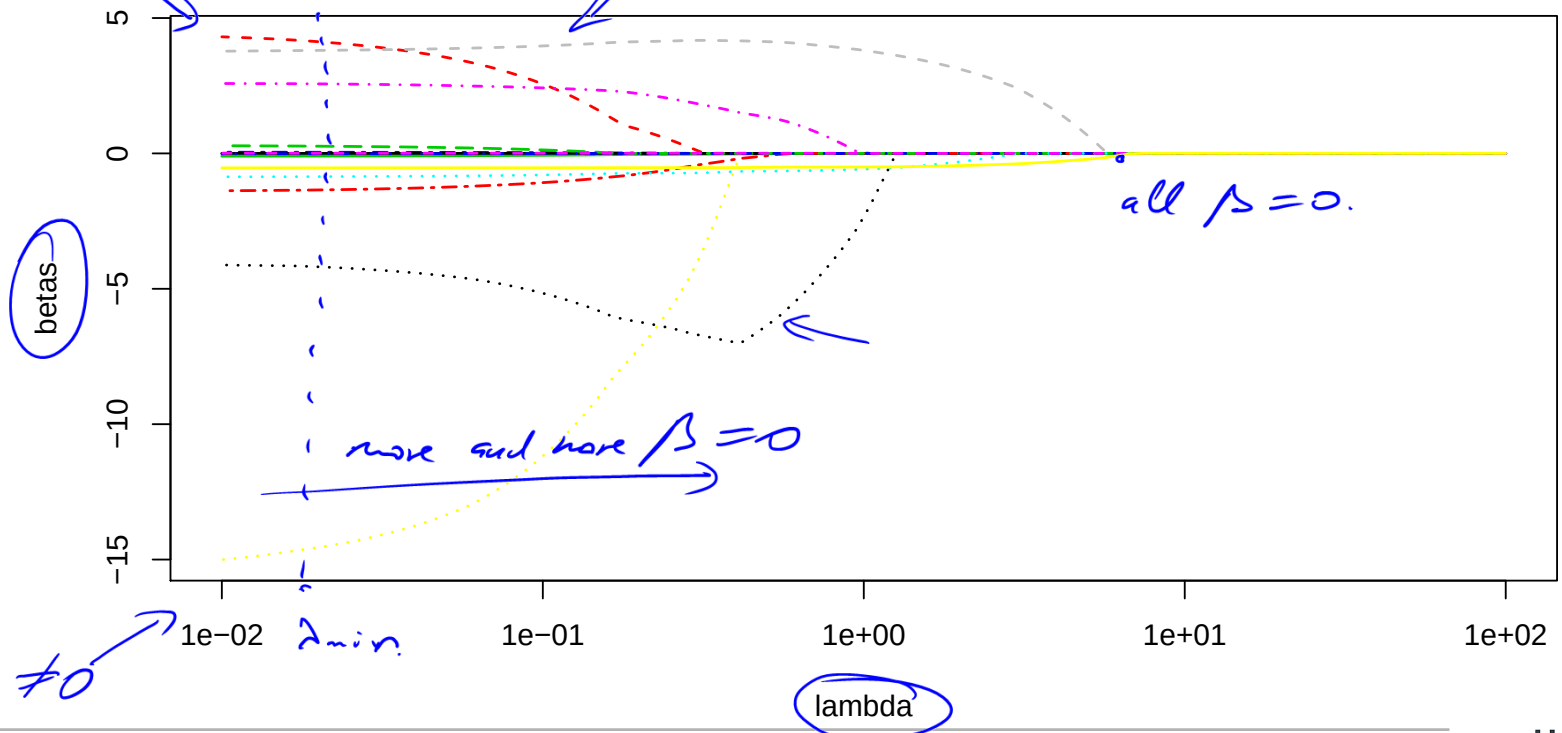
Example:

λ - small $\Rightarrow$ little penalty factor ($\lambda = 0 \Rightarrow$ OLS)
λ - large $\Rightarrow$ large penalty for β's ($\lambda \to \infty \Rightarrow \beta_j = 0$)

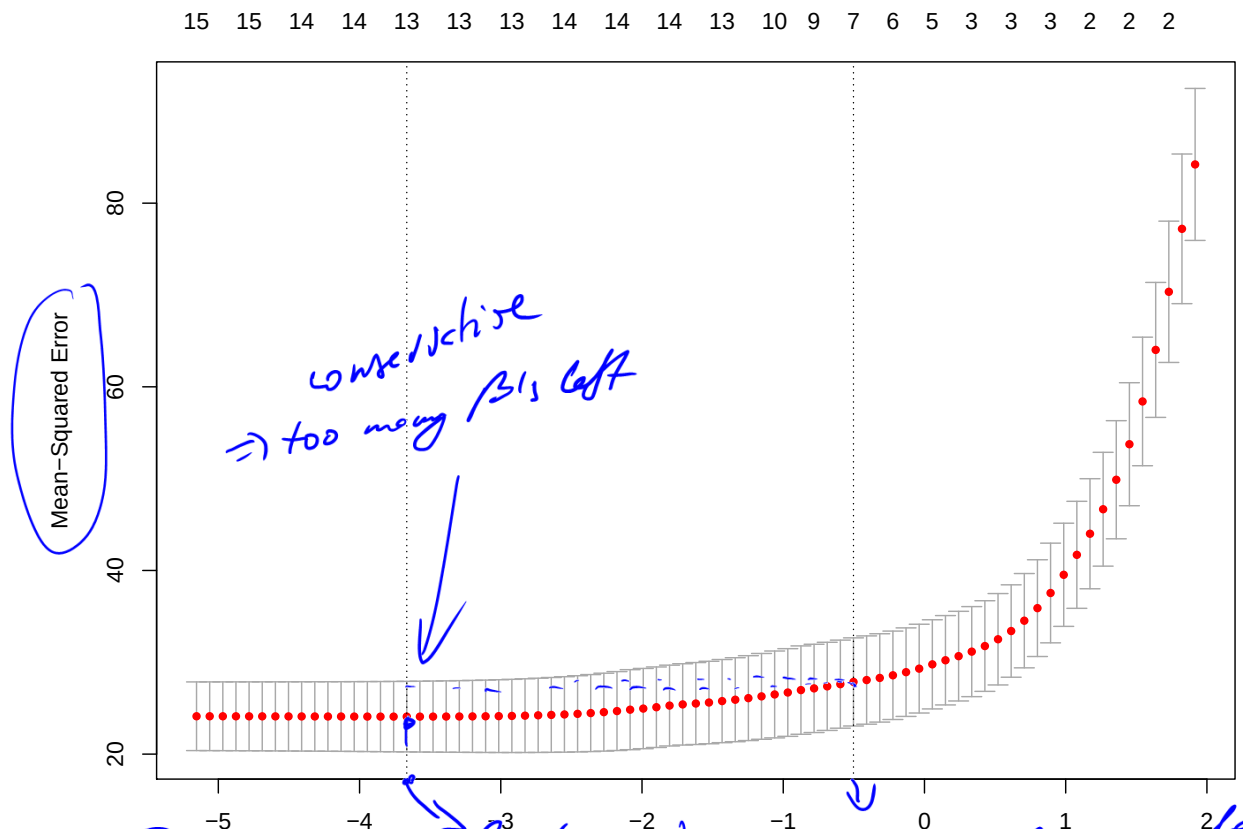
```
> grid = 10^seq(2,-2, length=100)
> lasso = glmnet(X, y.boston, alpha=1, lambda=grid);
> plot(lasso$beta)
```

alpha = 1 $\Rightarrow$ lasso
alpha = 2 $\Rightarrow$ ridge $\sum |\beta_j|^\alpha$

Boston housing data



```
> cv.lasso = cv.glmnet(X, y.boston, alpha=1);
> plot(cv.lasso)
> cv.lasso$lambda.min
[1] 0.0255856
```



```
> lasso.coef = predict(lasso, type="coefficients", s=cv.lasso$lambda.min);
> lasso.coef
16 x 1 sparse Matrix of class "dgCMatrix"
                         1
(Intercept) -4.392079e+02
lon         -4.250433e+00
lat          4.017435e+00
crim        -9.553123e-02
zn           4.149031e-02
indus        .
chas1        2.557345e+00
nox         -1.432740e+01
rm           3.816713e+00
age          .
dis         -1.329954e+00
rad          2.572257e-01
tax         -1.071061e-02
ptratio     -8.520927e-01
b            8.974756e-03
lstat       -5.326668e-01
```

type="coefficients" ... s=cv.lasso$lambda.min

↙ minimizes MSE

⇒ set to 0

⇒ set to 0.

estimation and feature selection in one step!