Validation

- Standard parametric methods
- Monte-Carlo simulation
- Fisher’s method of randomization
- Bootstrap methods

Monte-Carlo simulation

- Postulate statistical model with some free parameters to be estimated from data
- Estimate model parameters from real data using standard methods
- Generate new data sets with estimated parameters, by adding to analytical model randomly generated errors with assumed distribution
- Repeat the estimation procedure for every newly generated data set
- Compute ratio of ”sucessful” (do they show same period etc.) realizations to get significance of the solution
- If needed, compute standard deviations for parameters using gathered statistics

Problems with Monte-Carlo

- Only small set of analytical models is available
- Model can be inadequate (real data is more complex than conceived)
- Generation of realizations can be very complicted (it is hard to model all inherent in data correlations)

Fisher’s method of randomization

A randomization test is based on the idea that the observed arrangement of the data can be considered one of many possible arrangements, all of which are equally likely under the null hypothesis. Given a test statistic that is sensitive to the alternative hypothesis, the statistic is evaluated for each distinguishable arrangement of the data.

- Generate data permutations
- Fit models to permuted data or perform parameter (period) search
- Compute ratio of ”sucessfull” permutations

Problems with randomization

- The number of permutations is huge even for the small data sets
- The subsample of permutations (if used to speed up computations) may be uneven
- Inherent randomness of observed data values is not taken into account

Bootstrap methods

Random sample from an unknown probability distribution on the real line

$$X_1, X_2, \ldots, X_n \sim F.$$

Observed values

$$X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$$

Sample mean

$$\hat{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Standard error

$$\hat{\sigma} = \Big[\sum_{i=1}^{n} \frac{(x_i - \hat{x})^2}{(n-1)}\Big]^{1/2}$$

This is an old way, no to bootstrap!

Bootstrap

There are situations in which the standard error of the point estimator is unknown. Usually, these are cases where the form of $\hat{\Theta}$ is complicated, and the standard expectation and variance operators are difficult to apply. A computer-intensive technique called the **bootstrap** that was developed in recent years can be used for this problem.

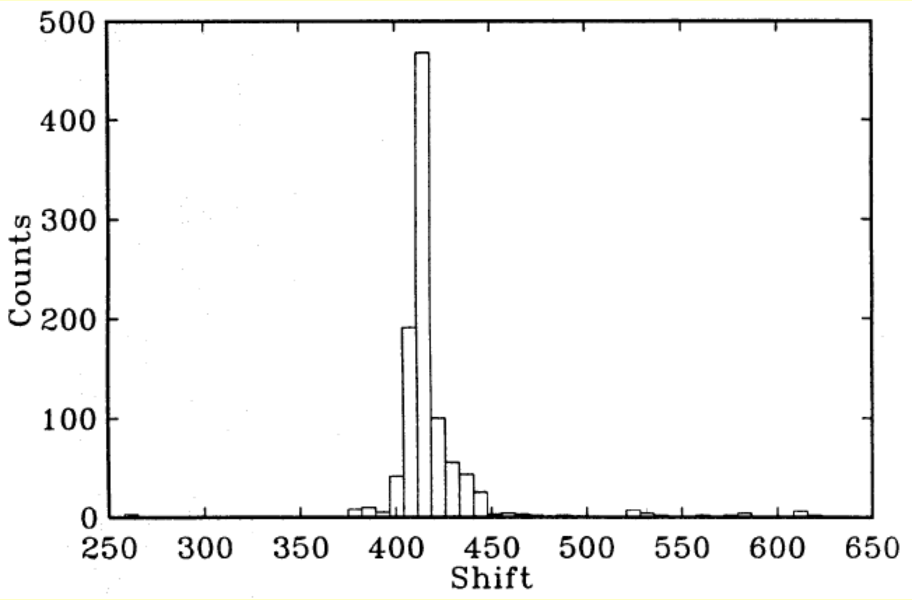
Suppose that we are sampling from a population that can be modeled by the probability distribution $f(x; \theta)$. The random sample results in data values $x_1, x_2, \ldots, x_n$ and we obtain $\hat{\theta}$ as the point estimate of θ. We would now use a computer to obtain *bootstrap samples* from the distribution $f(x; \hat{\theta})$, and for each of these samples we calculate the bootstrap estimate $\hat{\theta}^*$ of θ.

Bootstrap Sample	Observations	Bootstrap Estimate
1	$x_1^*, x_2^*, \ldots, x_n^*$	$\hat{\theta}_1^*$
2	$x_1^*, x_2^*, \ldots, x_n^*$	$\hat{\theta}_2^*$
$\vdots$	$\vdots$	$\vdots$
B	$x_1^*, x_2^*, \ldots, x_n^*$	$\hat{\theta}_B^*$

Usually $B = 100$ or 200 of these bootstrap samples are taken. Let $\overline{\theta}^* = (1/B)\sum_{i=1}^{B}\hat{\theta}_i^*$ be the sample mean of the bootstrap estimates. The bootstrap estimate of the standard error of $\hat{\Theta}$ is just the sample standard deviation of the $\hat{\theta}_i^*$, or

$$s_{\hat{\Theta}} = \sqrt{\frac{\sum_{i=1}^{B}(\hat{\theta}_i^* - \overline{\theta}^*)^2}{B - 1}} \tag{S7-1}$$

In the bootstrap literature, $B - 1$ in Equation S7-1 is often replaced by B. However, for the large values usually employed for B, there is little difference in the estimate produced for $s_{\hat{\Theta}}$.



Typical distribution of time delay estimates from bootstrap runs

Median filter and bootstrap

Let $f(t_i), i = 1, \ldots, N$ be a time series, M the filter length ($M \ll N$) and δ the maximum distance between the two observations ($\delta \ll t_n - t_1$) to be considered as nearby. Then for every observation $f(t_i)$ its filtered value $\hat{f}(t_i)$ is defined as the *median* value of the longest subsequence $f(t_{i-n}), n = -L/2, \ldots, L/2$, $L \leq M$ of the original data set which is centered around the point under discussion and which does not contain gaps longer than δ. This filter is extremely conservative since it leaves single observations untouched while measurement errors are suppressed for groups of nearby observations. Note that groups with less than M observations will be filtered with shorter filter lengths.

We applied an adaptive median filter to smooth the optimally combined light curve and then reshuffled the computed residuals 1000 times to generate standard bootstrap samples (Efron & Tibshirany 1986). The shifts obtained from generated samples were then used to calculate the error bars. This procedure gives reasonable bounds to the estimation errors in the case when the postulated model is indeed true. In our case the important point is that both light curves of the quasar are considered as two replicas of the one and the same original source. In reality the simple picture is spoiled by the strong microlensing event in the first part of the observational sequence. Happily enough, the second (unspoiled) part of the data set gave even sharper bounds (see 3.3).

Example: one-dimensional smoothing

(N=50 training data)

Fit a cubic spline with 3 knots:

$$\mu(x) = E(Y|X=x) = \sum_{j=1}^{7} \beta_j h_j(x) = h(x)^T \beta$$

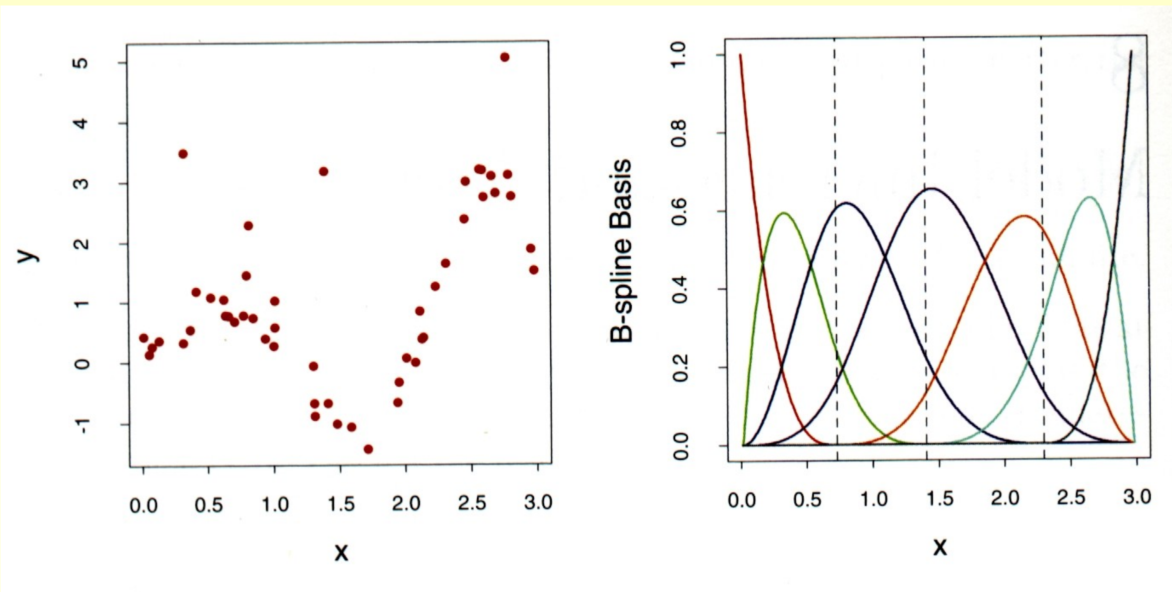
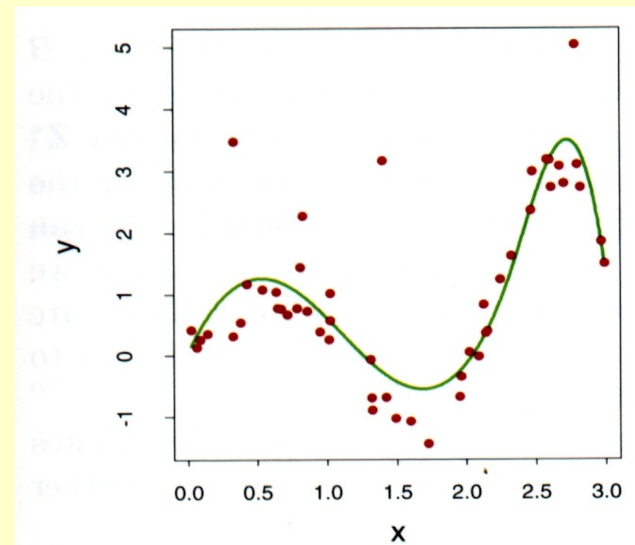


FIGURE 8.1. *Left panel: data for smoothing example. Right panel: set of seven B-spline basis functions. The broken vertical lines indicate the placement of the three knots.*



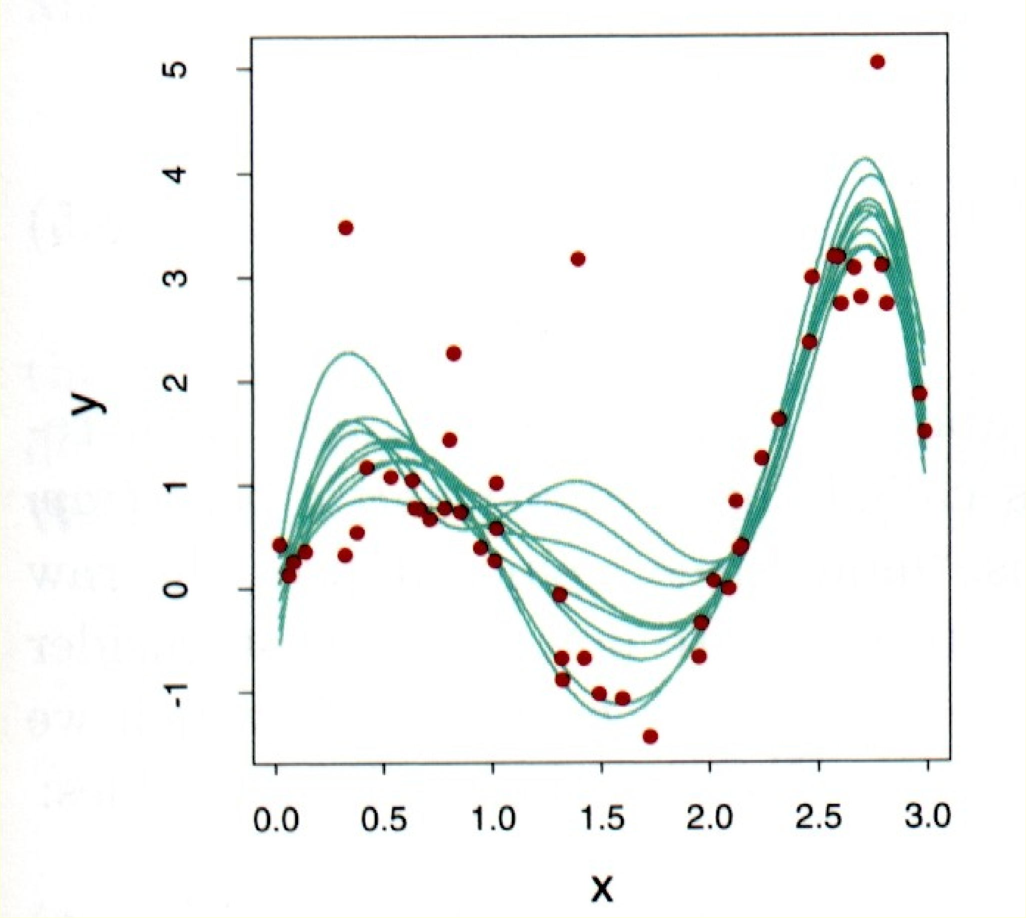
Bootstrap

Repeat B=200 times:

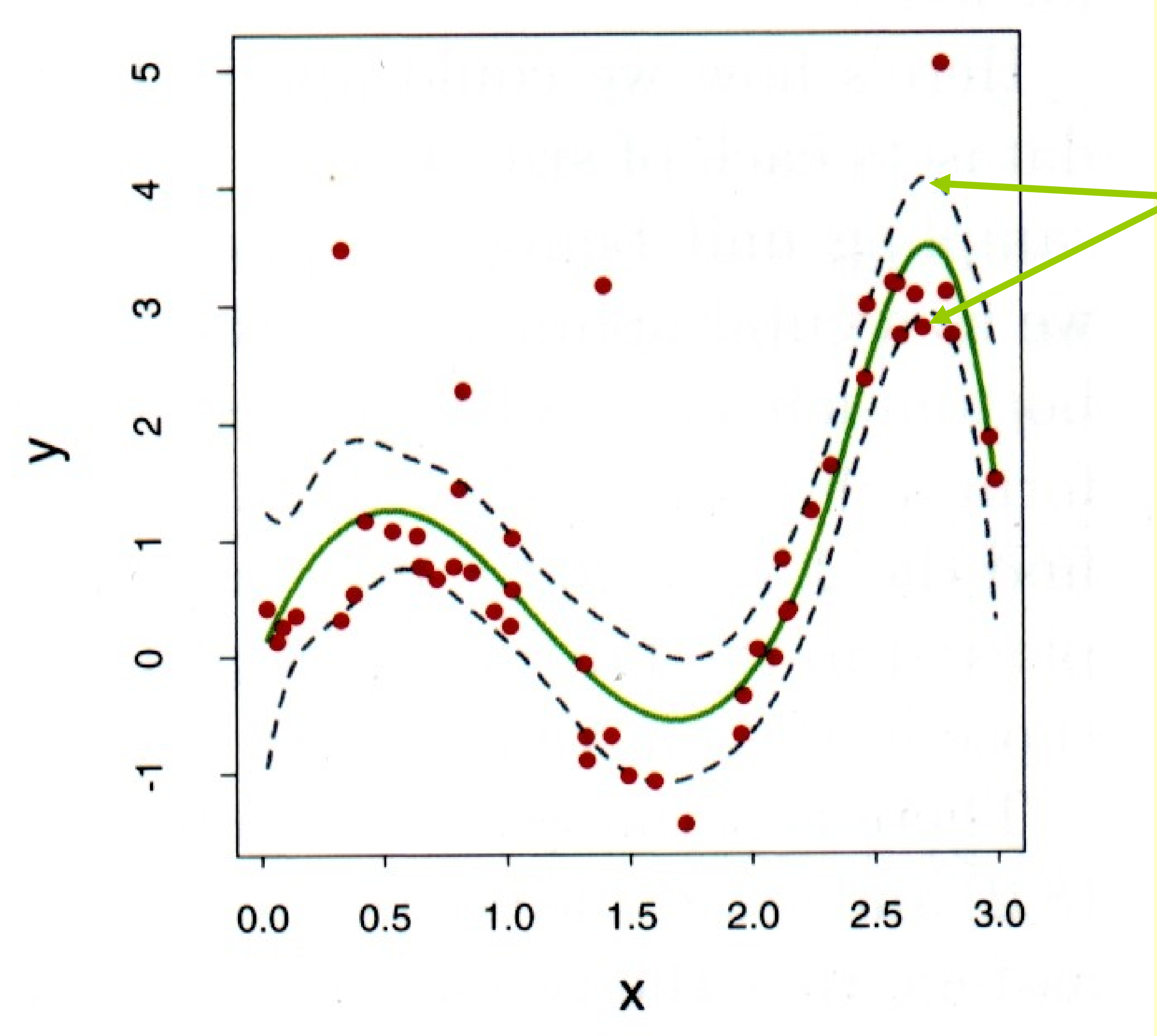
- draw a dataset of N=50 with replacement from the training data $z_i=(x_i,y_i)$
- fit a cubic spline

Construct a 95% pointwise confidence interval:

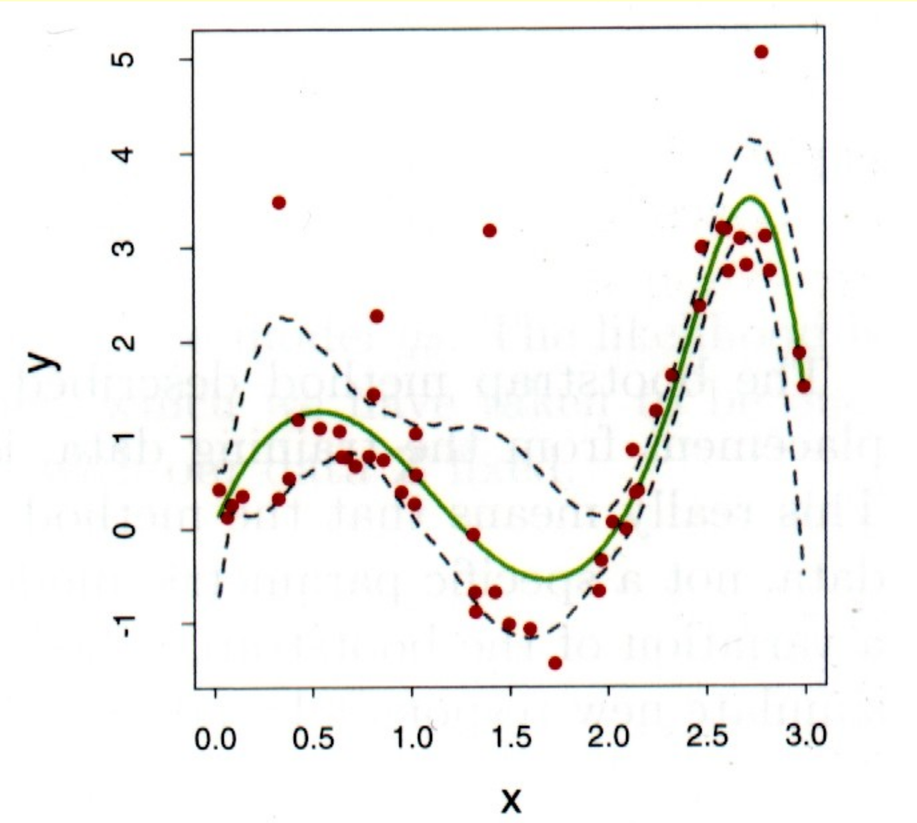
At each x_i compute the mean and find the 2,5% and 97,5% percentiles

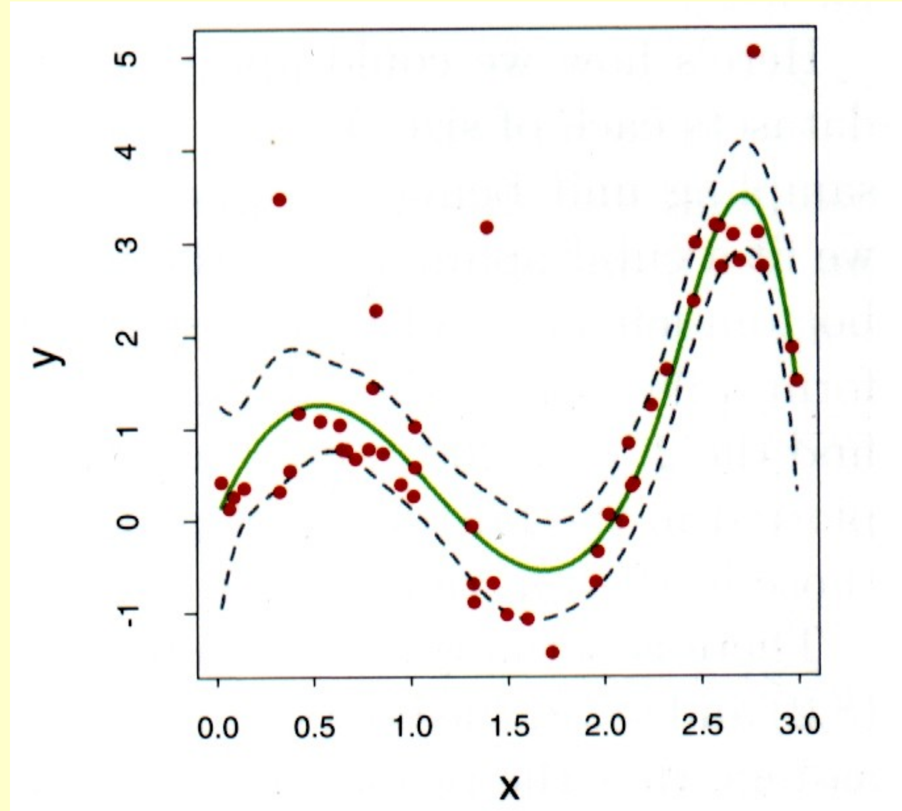


Compute Least Squares estimates



Normal prediction errors





Least squares   bootstrap

But bootstrap must be trusted more because of lesser amount of preliminary assumptions!