STAT 212: Principles of Statistics II

Lecture Notes: Chapter 1 (Part B)

More General Regression

Sharmistha Guha

Dept. of Statistics

Texas A&M University

Fall 2022

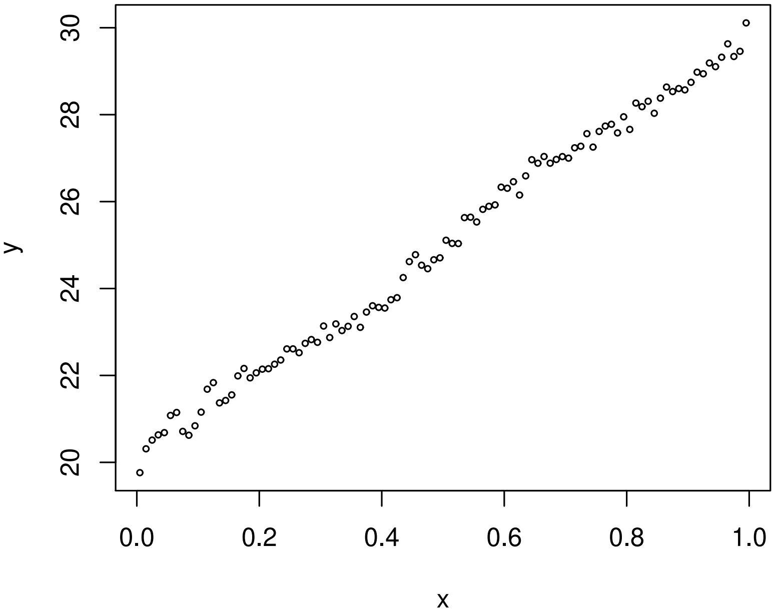
More General Regression Analyses

Eventually we'll study the case where we have more than one predictor/covariate/indep. var.

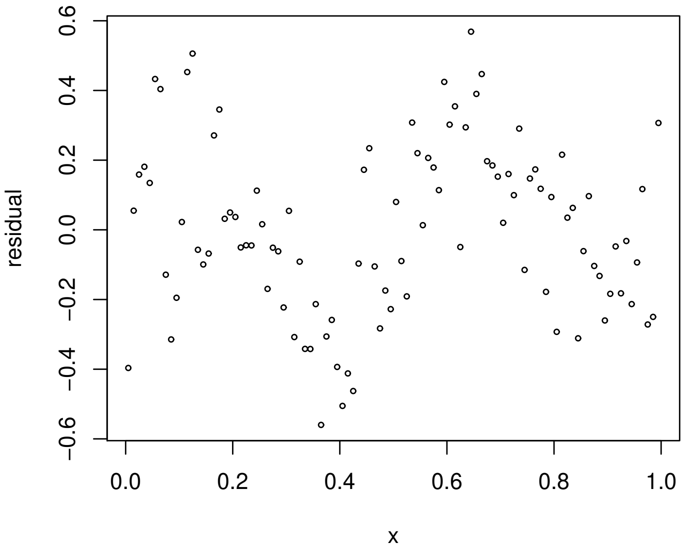
But initially we discuss a way of handling the one-variable case when a straight line fit (i.e. a linear regression model) isn't adequate.

Example 5: *Spotting nonlinearity via residuals*

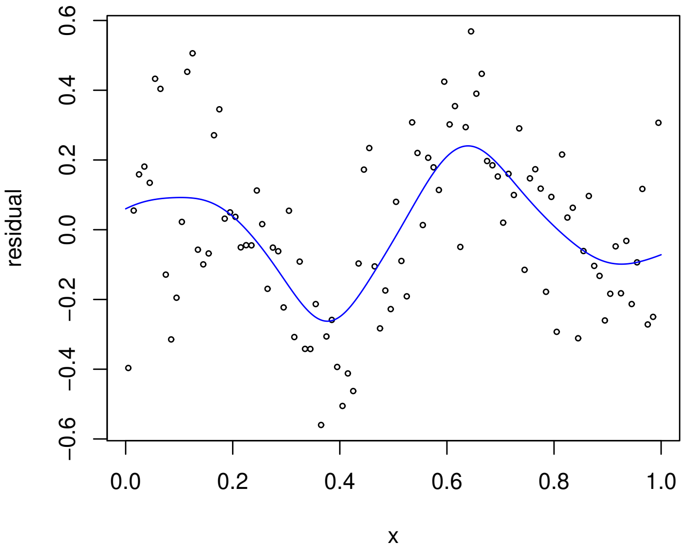
Simulated Data



Residuals from a straight line fitted to data on previous page



Residuals and smoothed residuals



The data for pg. 46N-47N were generated from the following model:

$$Y_j = 20 + 10x_j + 0.25\sin(4\pi x_j) + \epsilon_j,$$

$j = 1, \ldots, 100$, where each ϵ_j is normally distributed with mean 0 and std. deviation 0.2.

Previously our model was:

$$Y = \beta_0 + \beta_1 x + \epsilon.$$

More generally, we could assume:

$$Y = f(x) + \epsilon,$$

where f *is an arbitrary function*.

However, when doing a statistical analysis we must limit the set of candidate models.

Otherwise there would be too many parameters to estimate.

Principle of parsimony or *Occam's razor:*

> "The best explanation of a phenomenon is the simplest one that fits the facts."

Statistical analog of Occam's razor:

> "Use the simplest model, i.e., the one with fewest parameters, that still fits the data adequately."

A good way to obtain a "flexible" class of models while limiting the number of parameters is to use polynomials (i.e. $f(x)$ is a polynomial).

Taylor's theorem tells us that a "smooth" function can be approximated arbitrarily well by a polynomial of sufficiently high degree.

We will assume that:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k + \epsilon.$$

As before $\epsilon \sim N(0, \sigma^2)$.

Given the data $(x_1, y_1), \ldots, (x_n, y_n)$, we can estimate the polynomial coefficients using least squares (the principle stays same as before).

$$f(b_0, b_1, \ldots, b_k) = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} b_j x_i^j \right)^2$$

Choose $b_0, \ldots, b_k$ to minimize $f(b_0, b_1, \ldots, b_k)$.

$$\frac{\partial f(b_0, \ldots, b_k)}{\partial b_j} = 0, \quad j = 0, 1, \ldots, k$$

This leads to the set of equations on the next page which are the so-called *normal equations*.

$$\sum_{j=0}^{k} b_j \sum_{i=1}^{n} x_i^j = \sum_{i=1}^{n} y_i$$

$$\sum_{j=0}^{k} b_j \sum_{i=1}^{n} x_i^{j+1} = \sum_{i=1}^{n} x_i y_i$$

$$\vdots$$

$$\sum_{j=0}^{k} b_j \sum_{i=1}^{n} x_i^{j+k} = \sum_{i=1}^{n} x_i^k y_i$$

These have the form

$$a_{\ell 0} b_0 + a_{\ell 1} b_1 + \cdots + a_{\ell k} b_k = c_\ell, \quad \ell = 0, \ldots, k,$$

and are thus *linear equations*.

Linear equations are easy to solve, which is a big plus for polynomial regression.

We may derive the solution by writing the equations in matrix form.

Notation: If $\boldsymbol{A}$ is a $p \times q$ matrix, then $\boldsymbol{A}^T$ denotes the transpose of $\boldsymbol{A}$. $\boldsymbol{A}^T$ is a $q \times p$ matrix whose columns are the same as the rows of $\boldsymbol{A}$.

Define the following matrices:

$$\boldsymbol{b} = [b_0 \ b_1 \ \cdots \ b_k]^T$$

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ & & \vdots & & \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix}$$

$$\boldsymbol{y} = [y_1 \ y_2 \ \cdots \ y_n]^T$$

The normal equations in matrix form are:

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b} = \boldsymbol{X}^T\boldsymbol{y}.$$

Assuming that $\boldsymbol{X}^T\boldsymbol{X}$ is invertible, the soln. is:

$$\boldsymbol{b} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \widehat{\boldsymbol{\beta}}.$$

The components of $\widehat{\boldsymbol{\beta}}$ are the least squares estimates of $\beta_0, \beta_1, \ldots, \beta_k$.

For deeper details/understanding of these matrix forms, see the notes for Lab Session 3.

Estimating σ^2

Predicted values:

$$\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i + \cdots + \widehat{\beta}_k x_i^k, \quad i = 1, \ldots, n$$

Residuals:

$$e_i = y_i - \widehat{y}_i, \quad i = 1, \ldots, n$$

$$SSE = \sum_{i=1}^{n} e_i^2$$

We estimate σ^2 by:

$$\widehat{\sigma}^2 = MSE = \frac{SSE}{n-k-1}.$$

As in the straight line case: $SST = SSR + SSE$,

with $\quad SST = \sum_{i=1}^{n} (y_i - \bar{y})^2, \quad SSR = \sum_{i=1}^{n} (\widehat{y}_i - \bar{y})^2$

and $\quad R^2 = \dfrac{SSR}{SST} = 1 - \dfrac{SSE}{SST}.$

R^2 is the *fraction of the total variation in the y_i's explained by the polynomial model.*

Using R to fit polynomials

There are two ways to fit, say, a third degree polynomial:

- `lm(y~poly(x,3,raw=T))`, or equivalently `lm(y~x+I(x^2)+I(x^3))`

Both methods produce *exactly the same predicted values and residuals.*

The commands

```
fit=lm(y~poly(x,3,raw=T))
summary(lm(y~poly(x,3,raw=T)))
```

provide us with similar looking output as in the straight line case. Actually all prior `R` commands continue to apply (pg. 15N, 20N, 38N).

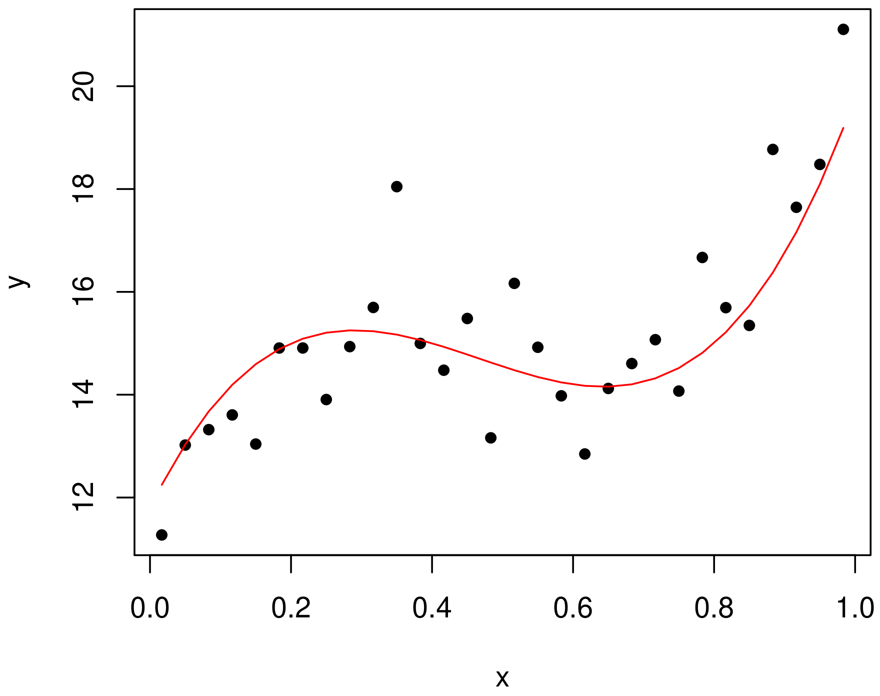
Example 6: Fitting different degree polynomials to a data set

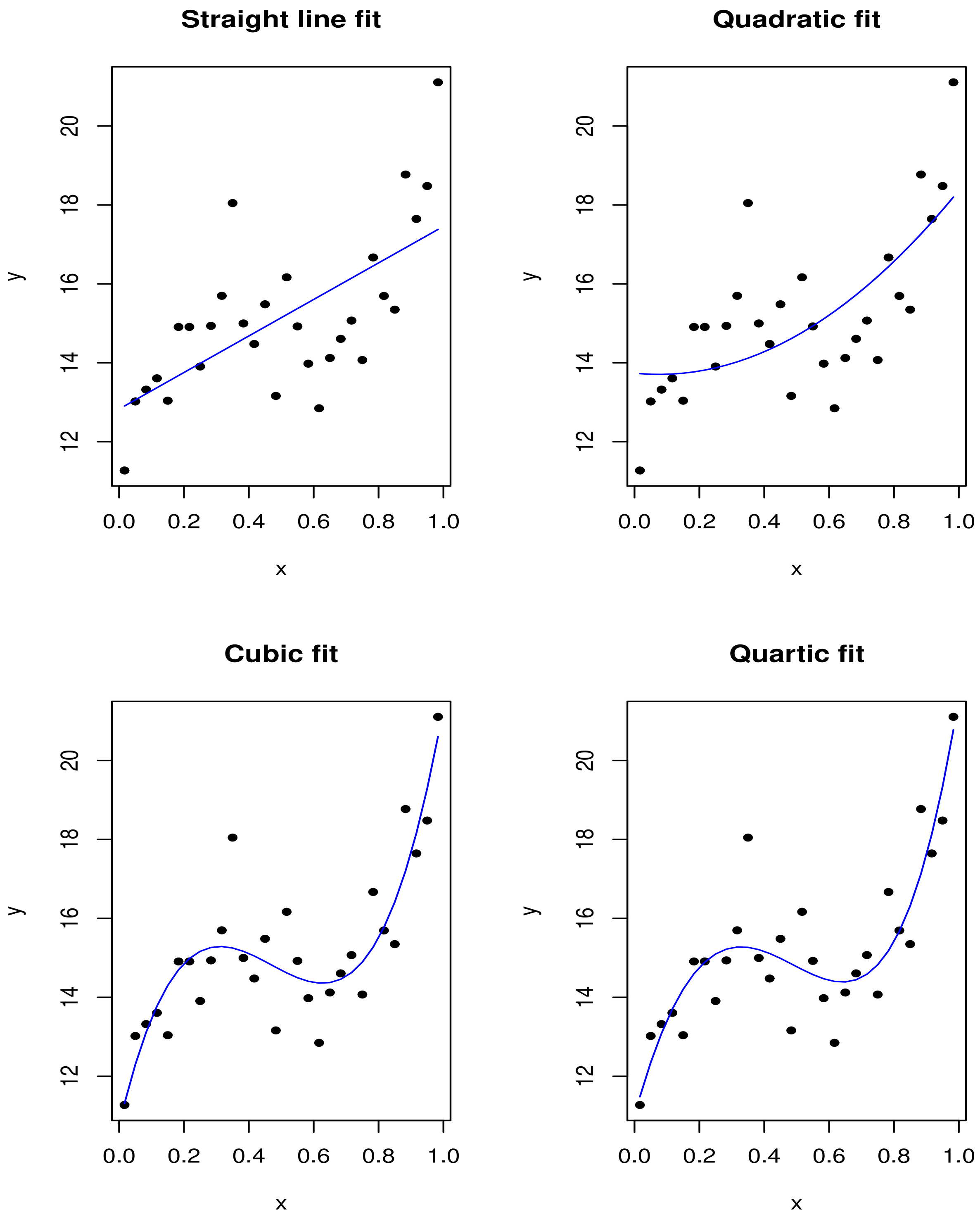
Data were simulated from the model

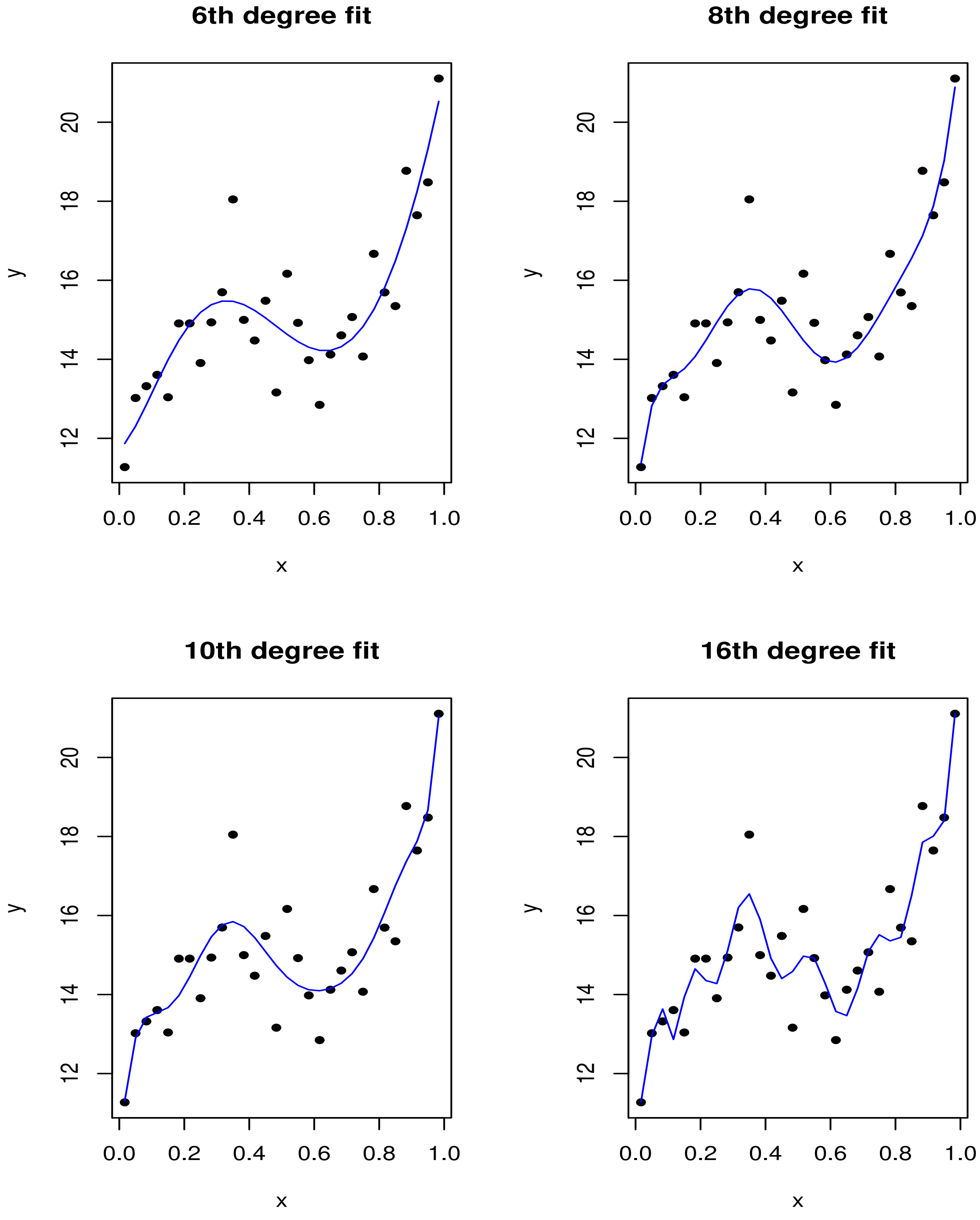
$$Y_i = 11.8 + 28x_i - 70x_i^2 + 50x_i^3 + \epsilon_i,$$

where $x_i = (i - 1/2)/30$, $i = 1, \ldots, 30$, and $\epsilon_i \sim N(0, 1)$.

Simulated data and true cubic polynomial




Straight line fit
Quadratic fit
Cubic fit
Quartic fit
y
x
12
14
16
18
20
0.0
0.2
0.4
0.6
0.8
1.0



6th degree fit

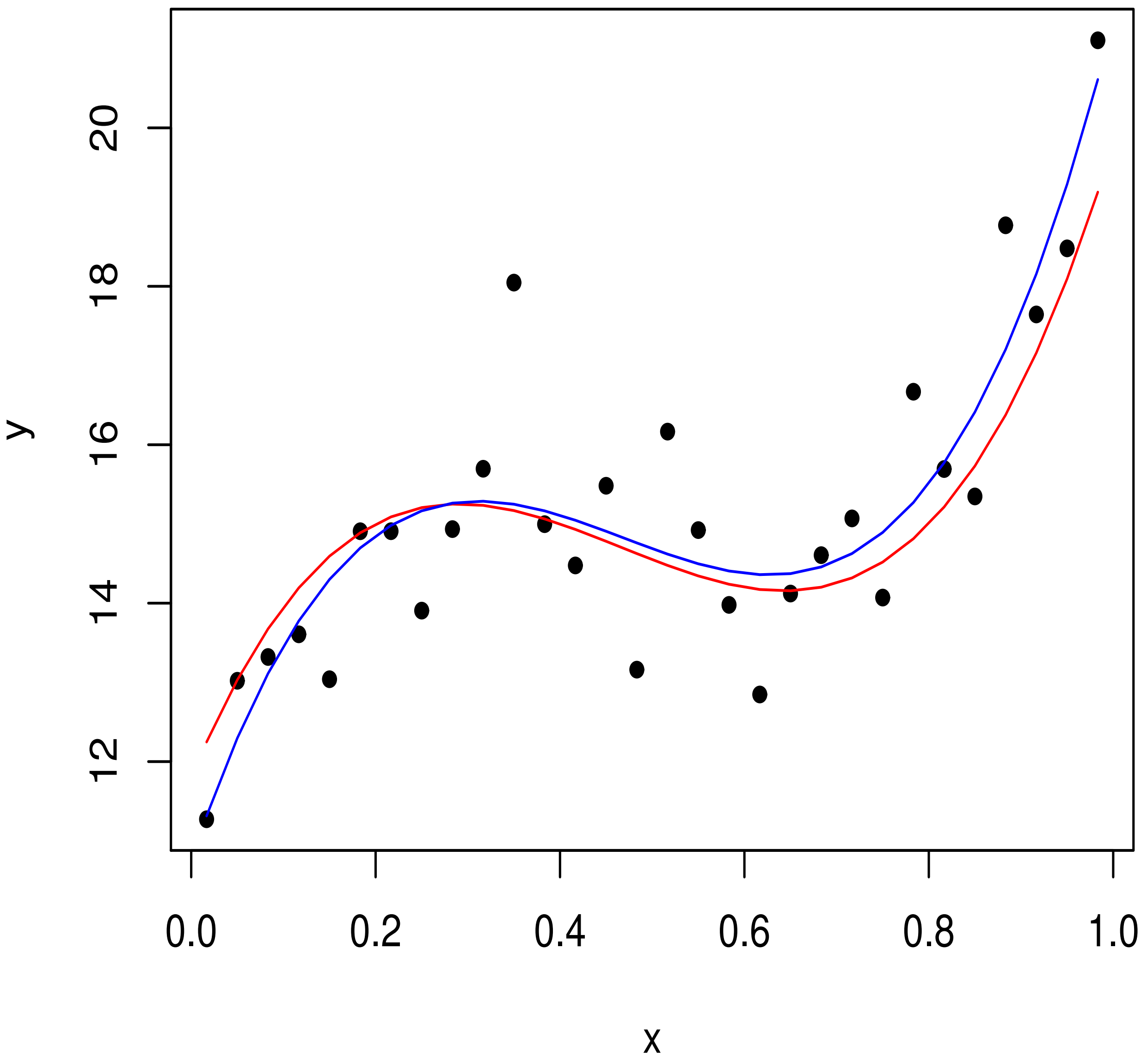
8th degree fit

10th degree fit

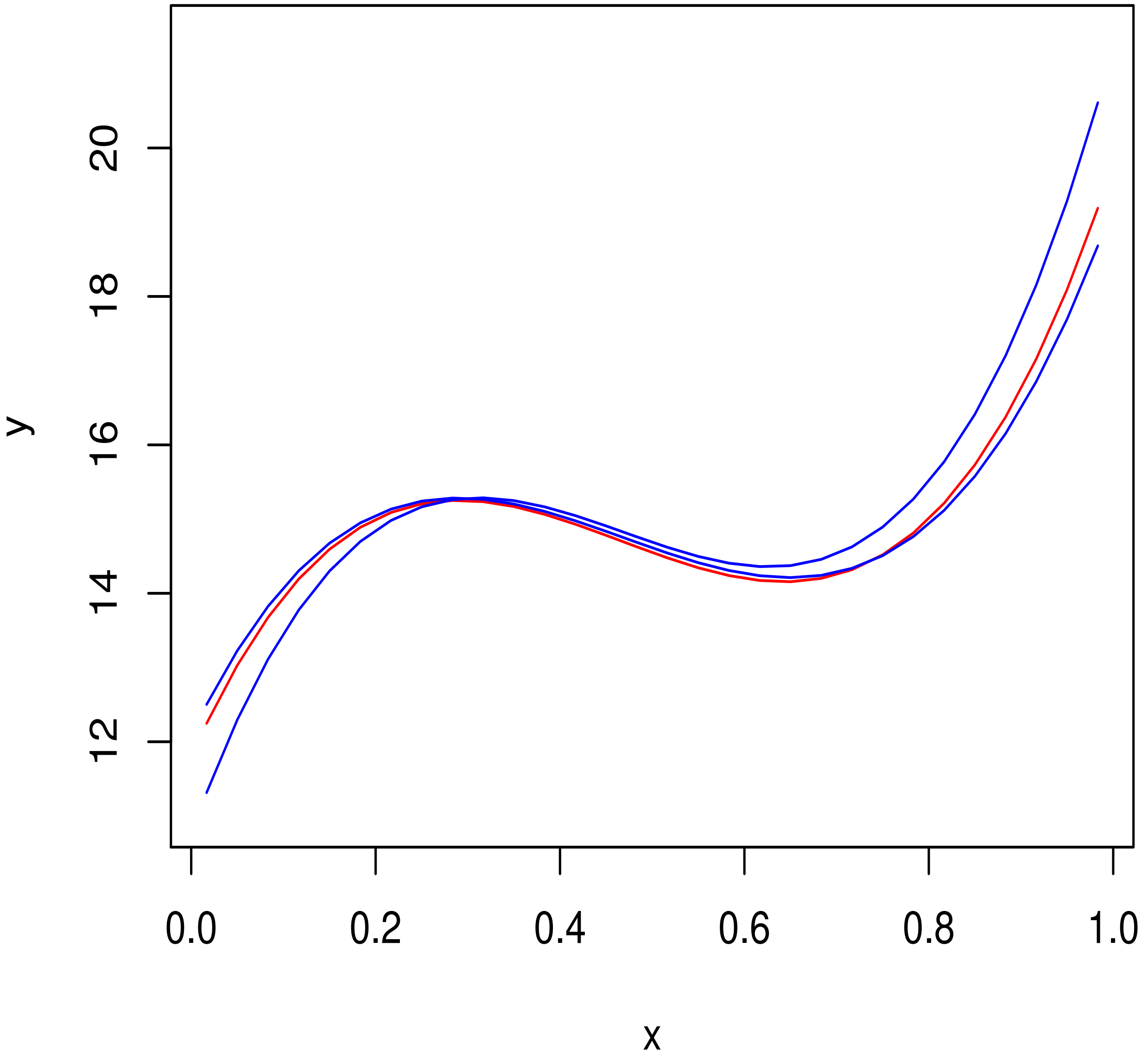
16th degree fit

Red: true curve

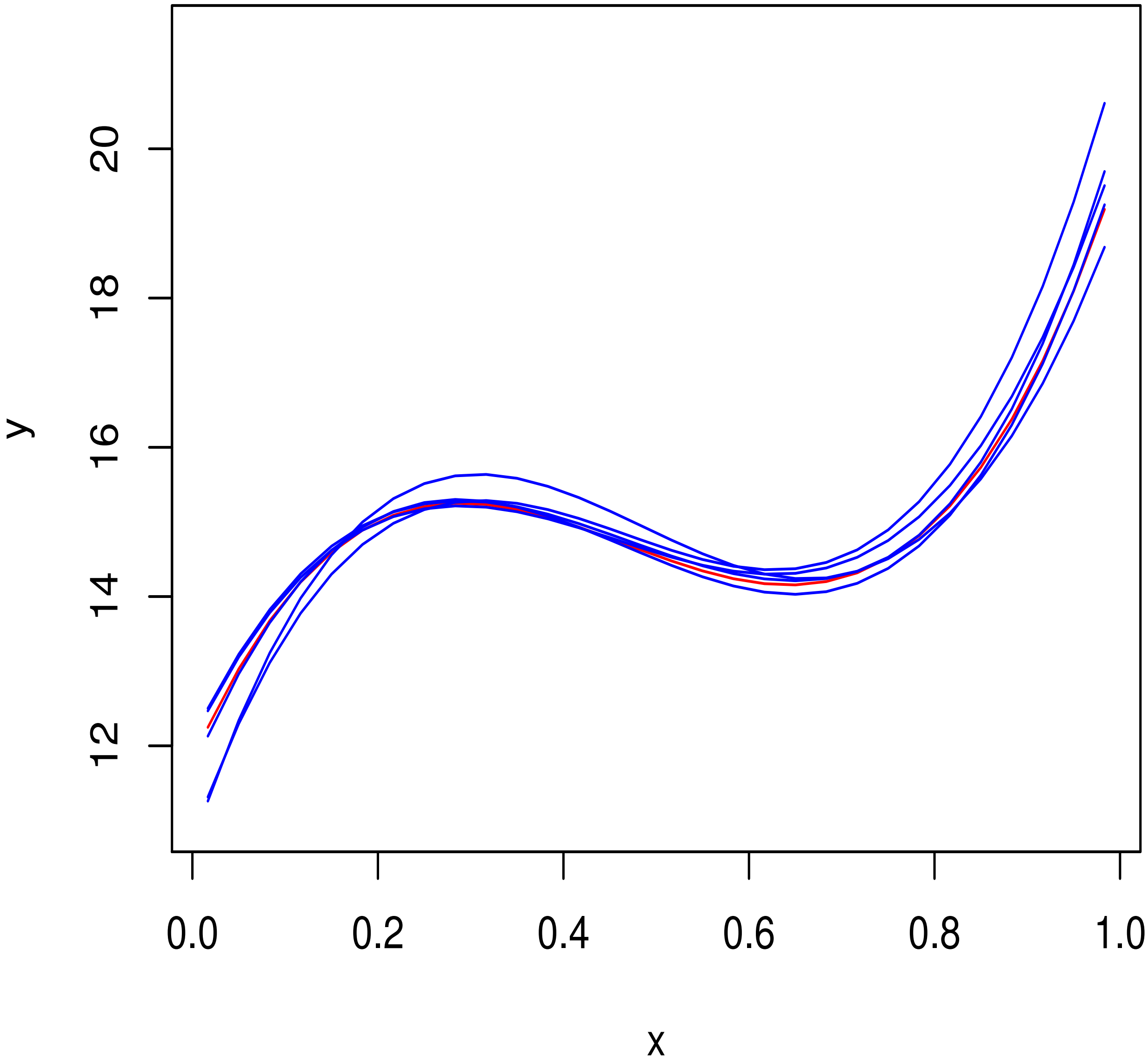
Blue: cubic fit



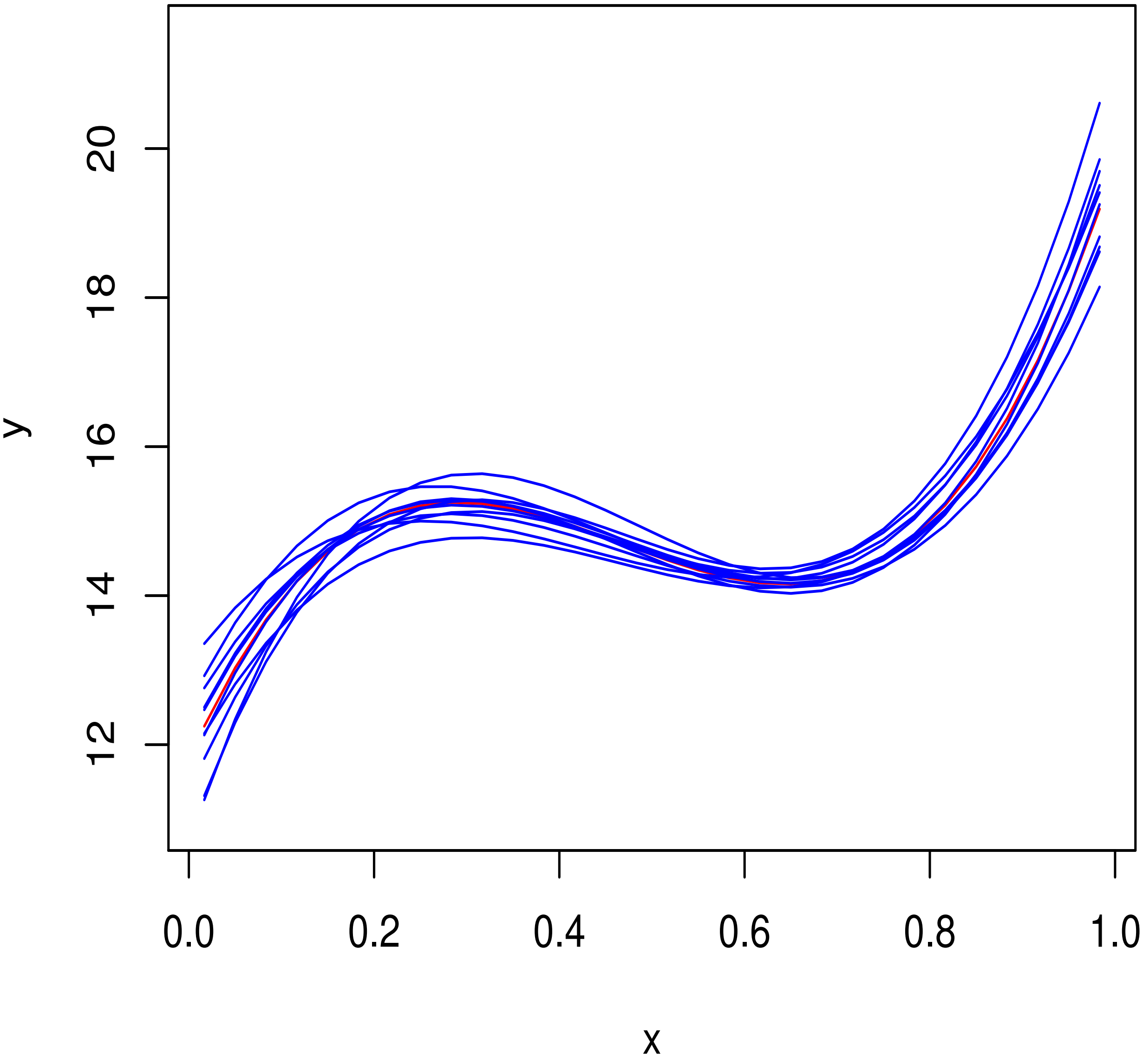
Fitted curves from two different data sets



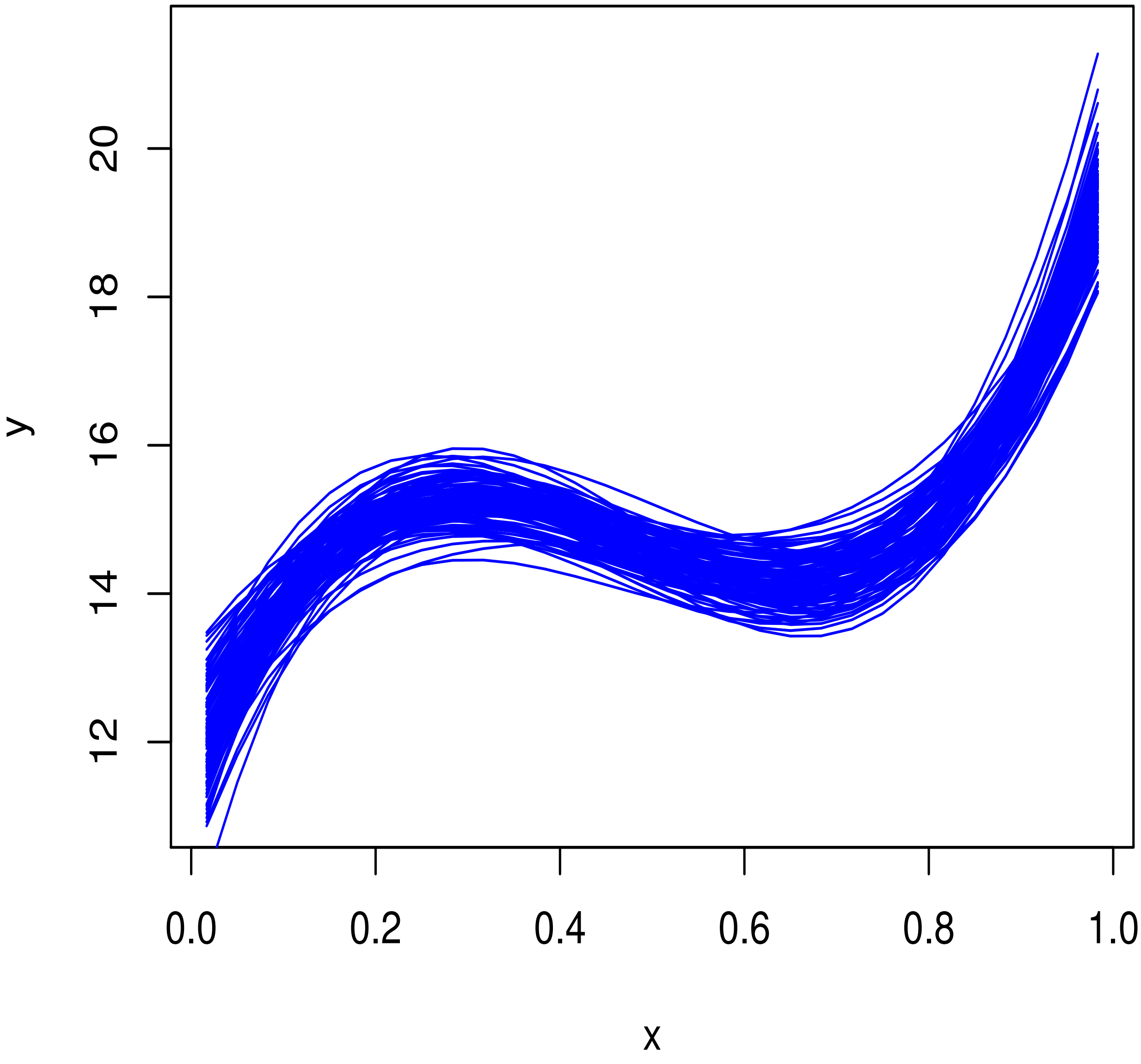
. . . *from five data sets*



. . . from ten data sets



. . . and from 100 data sets



Question: How do you do these with real data?
Answer: bootstrap; see Supplementary Notes.

Adjusted R^2

When using polynomials to estimate a regression function, we won't always know beforehand (i.e., before data collection) what a good polynomial degree is.

One might be tempted to say "Use the degree that gives the highest R^2 value."

This turns out to be unreliable since R^2 is guaranteed to be an increasing function of polynomial degree.

Let R_k^2 be the R^2 value when a kth degree polynomial is fitted to our data. Then

$$R_1^2 < R_2^2 < \cdots < R_{n-2}^2 < R_{n-1}^2 = 1.$$

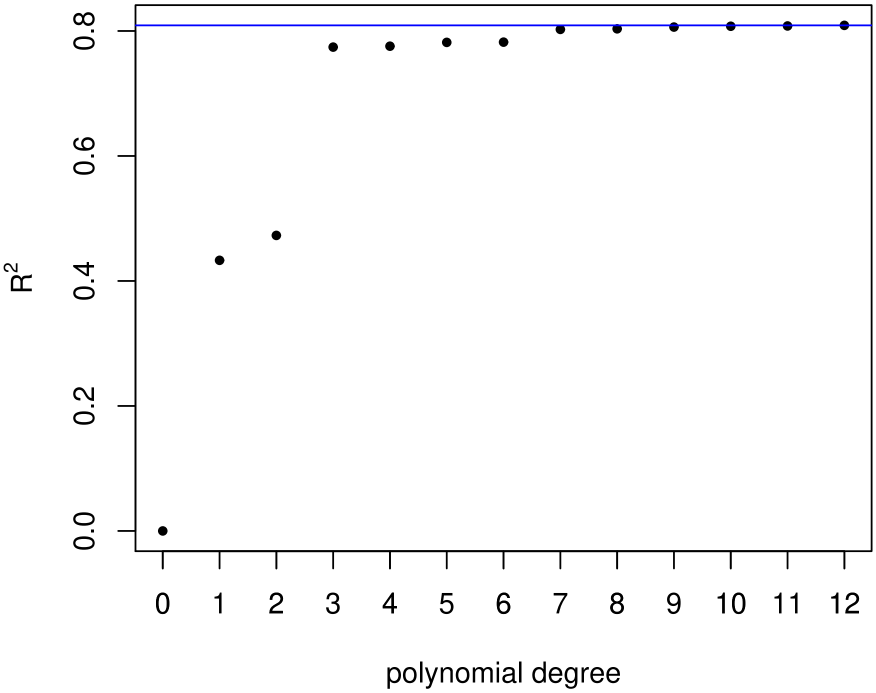
So, maximizing R^2 will **not** help us to act in accord with *Occam's razor*.

One solution: Use *adjusted R^2*, denoted $R^2_{\text{adj,k}}$.

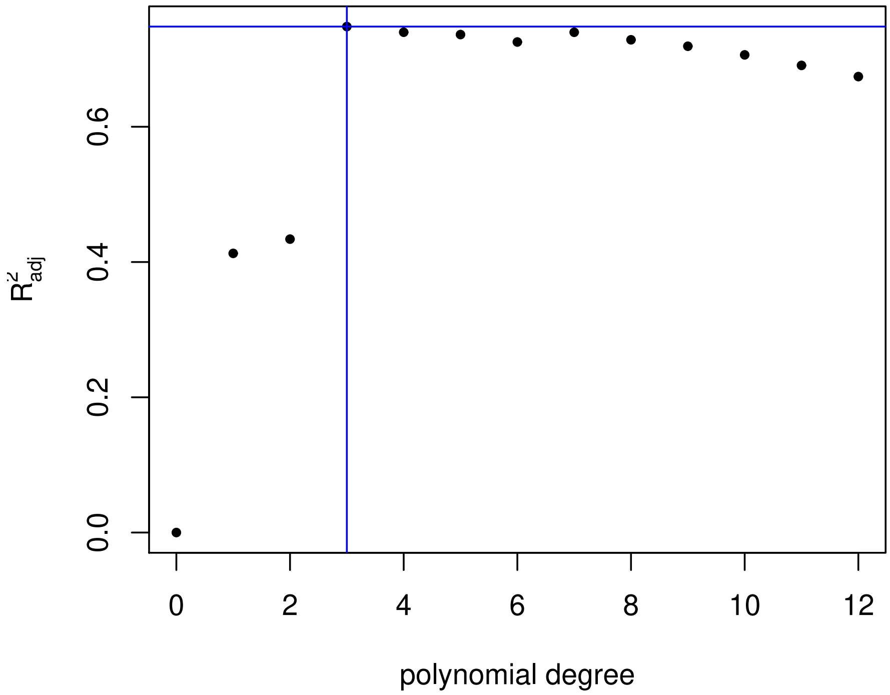
$$R^2_{\text{adj,k}} = \frac{(n-1)R^2_k - k}{n-1-k}$$
$$= R^2_k - \frac{k}{(n-1-k)}(1 - R^2_k).$$

So $R^2_{\text{adj,k}}$ penalizes R^2_k and the penalty increases with k. Choose k to maximize $R^2_{\text{adj},k}$. *The higher degree polynomials are penalized more.*

Unadjusted R^2 for data on pg. 55N



Adjusted R^2 for data on pg. 55N



So, R^2_{adj} worked very well in this example since it is maximized at 3, which matches the true polynomial degree.

Two other methods of selecting polynomial degree are called AIC (Akaike Information Criteria) and BIC (Bayes Information Criteria).

Using R to get AIC, BIC and Adjusted R^2

Suppose you use the command

```
fit=lm(y~poly(x,3,raw=T))
```

to fit a third degree polynomial model. You may obtain the values of AIC, BIC and R^2_{adj} as follows:

- AIC: `AIC(fit)`
- BIC: `BIC(fit)`
- Adjusted R^2: `summary(fit)$adj`

Choose polynomial degree to *minimize* AIC (and likewise for BIC).

How well do the three methods work?

I generated 1000 data sets from the third degree polynomial model on pg. 55N.

For each of the 1000 data sets, I determined the degree that was selected by AIC, BIC and adjusted R^2, where degrees from 1 through 12 were considered.

Percentage of 1000 cases where selectors chose degree k

k	AIC	BIC	R^2_{adj}
1	0.1	0.2	0
2	0	0	0
3	52.5	84.0	23.1
4	9.1	6.6	8.1
5	6.7	4.2	7.1
6	4.5	1.9	6.6
7	3.1	0.5	5.5
8	3.9	0.5	7.4
9	3.9	0.7	8.6
10	3.7	0.4	7.2
11	4.8	0.5	9.2
12	7.7	0.5	17.2

It's clear that BIC did the best job of selecting the correct degree, while AIC was second best.

Adjusted R^2 was a poor third. It has a stronger tendency to greatly *overestimate* the right degree than do AIC and BIC.

What happens if n is increased here? All results get more precise and with higher chances of choosing the correct degree; see Supp. Notes.

<u>FYI</u>, one accepted definition for AIC and BIC:

Let SSE_k denote the SSE for the model fitted with the k^{th} degree polynomial. Then, AIC and BIC for this k^{th} degree model are given by:

$$\text{AIC}_k = n \log(SSE_k) + 2k \quad \text{and}$$

$$\text{BIC}_k = n \log(SSE_k) + k \log(n).$$

Note that BIC penalizes more than AIC, and hence, tends to select smaller/simpler models.

Inference problems

Suppose we want to test a hypothesis about or construct a confidence interval for one of the β_is in our polynomial regression model.

For example, consider the model

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i.$$

This says that the polynomial degree is no more than 3, but possibly less than 3.

Interesting hypotheses for the cubic model are

$$H_0 : \beta_3 = 0 \quad \text{vs.} \quad H_a : \beta_3 \neq 0.$$

H_0 says we have a quadratic model.

H_a says we have a cubic model.

The standard error of $\widehat{\beta}_i$ is $\sigma\sqrt{c_{i+1}}$, where for any $j = 1, \ldots, k+1$,

$$c_j = j\text{th element on the diagonal of } (\boldsymbol{X}^T\boldsymbol{X})^{-1}.$$

When $\epsilon_1, \ldots, \epsilon_n$ are a random sample from the normal distribution with mean 0 and var. σ^2,

$$\frac{\widehat{\beta}_i - \beta_i}{\widehat{\sigma}\sqrt{c_{i+1}}} \sim t_{n-k-1}.$$

So, a $(1-\alpha)100\%$ confidence interval for β_i is:

$$\widehat{\beta}_i \pm t_{n-k-1;\alpha/2}\widehat{\sigma}\sqrt{c_{i+1}}.$$

Suppose we want to estimate the mean of Y at x-value x_0, or predict Y corresponding to an x-value of x_0.

The point estimate of the mean of Y at x_0 is:

$$\widehat{\beta}_0 + \widehat{\beta}_1 x_0 + \cdots + \widehat{\beta}_k x_0^k = \widehat{\mu}(x_0).$$

$(1-\alpha)100\%$ confidence interval for the mean response, i.e. mean of Y, at x_0:

$$\widehat{\mu}(x_0) \pm t_{n-k-1;\alpha/2}\,\widehat{\sigma}\sqrt{\boldsymbol{x}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}},$$

$$\text{where} \quad \boldsymbol{x}^T = (1 \;\; x_0 \;\; x_0^2 \;\; \cdots \;\; x_0^k).$$

A $(1-\alpha)100\%$ prediction interval for Y at x_0:

$$\widehat{\mu}(x_0) \pm t_{n-k-1;\alpha/2}\,\widehat{\sigma}\sqrt{1+\boldsymbol{x}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}}.$$

Inference for polynomial regression via R

All the commands we talked about in the straight line case (i.e. for simple linear regression) work exactly the same in polynomial regression.

Suppose we have fitted a third degree polynomial using the following command:

```
fit=lm(y~poly(x,3,raw=T))
```

Then the commands `summary`, `confint` and `predict` apply just as before.

We'll use the data in `WindSpeed.txt` to illustrate fitting a polynomial model using R.

A more detailed implementation will be discussed in the Lab Sessions. For a summary of the main analyses done here, see Supp. Notes.

Non-polynomial models

Suppose

$$Y_i = f(x_i; \theta_1, \ldots, \theta_k) + \epsilon_i, \quad i = 1, \ldots, n,$$

where the ϵ_i's satisfy our usual assumptions and the regression function $f(x; \theta_1, \ldots, \theta_k)$ is known except for the parameters $\theta_1, \ldots, \theta_k$.

For example,

$$f(x; \theta_1, \theta_2, \theta_3) = \exp(\theta_1 + \theta_2 x + \theta_3 x^2)$$

or

$$f(x; \theta_1, \theta_2) = \theta_1 \cos(2\pi\theta_2 x).$$

The choice of functional form would be based on knowledge of the particular problem.

Least squares can *still* be used to estimate the model parameters $\theta_1, \ldots, \theta_k$.

Least squares estimates are those values $\widehat{\theta}_1, \ldots,$ $\widehat{\theta}_k$ that minimize

$$\sum_{i=1}^{n} [y_i - f(x_i; a_1, \ldots, a_k)]^2$$

with respect to $a_1, \ldots, a_k$.

Finding least squares estimates is often more difficult for non-polynomial models. Usually the equations to be solved are *non-linear*.

Transforming variables

Sometimes a model can be turned into a simple linear model by transforming x and/or y.

For example, suppose

$$Y = \alpha e^{\beta x} \epsilon,$$

where ϵ is a positive-valued random variable. Here the error is *multiplicative* rather than *additive* like it was before.

Taking the log of Y gives

$$\log Y = \log \alpha + \beta x + \log \epsilon.$$

Now, defining

$$Y' = \log Y, \quad \beta_0 = \log \alpha, \quad \beta_1 = \beta$$

and $\epsilon' = \log \epsilon$, we get

$$Y' = \beta_0 + \beta_1 x + \epsilon',$$

which is just our old straight line model.

Similarly, if $Y = \alpha x^{\beta} \epsilon$ $(x > 0)$,

$$\log Y = \log \alpha + \beta \log x + \log \epsilon.$$

Taking $x' = \log x$, again we have our familiar straight line model.

For $Y = \alpha x^{\beta} \epsilon$ we have

$$\text{Var}(Y) = \alpha^2 x^{2\beta} \text{Var}(\epsilon),$$

and so even if ϵ has the same variance for all x, Var(Y) will *increase as x increases*. On the other hand,

$$\text{Var}(\log Y) = \text{Var}(\log \epsilon),$$

which is the same for all x if the probability distribution of ϵ doesn't depend on x.

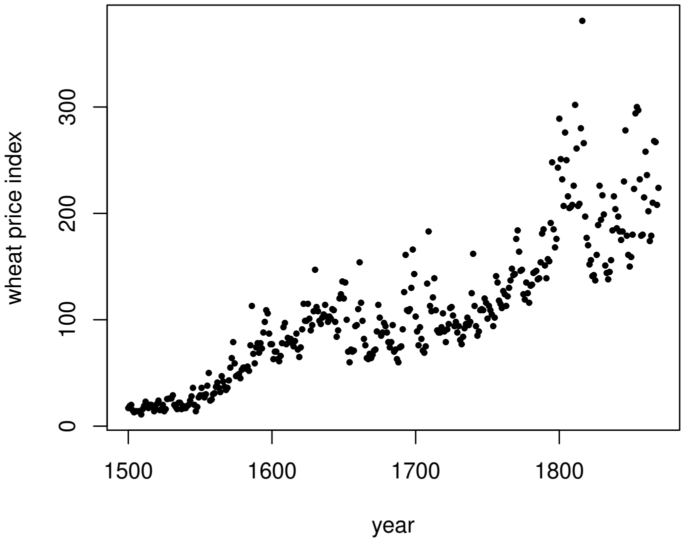
When the variance of the response appears to be increasing or decreasing with x, this is a tipoff that taking logs may be advisable.

Example 7: *European wheat prices*

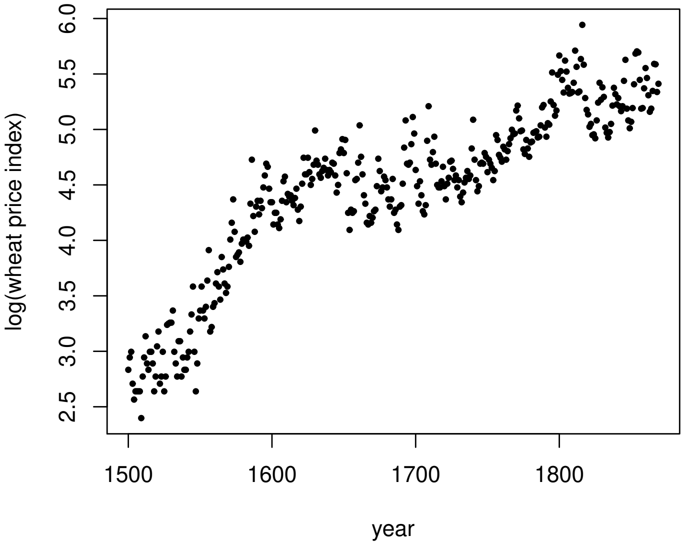
W.H. Beveridge was a British economist who lived from 1879 to 1963. He devised an index of wheat prices in western and central Europe for the period from 1500 to 1869.

The series of data consists of averages of wheat prices from nearly 50 places, and is of interest to economic historians.

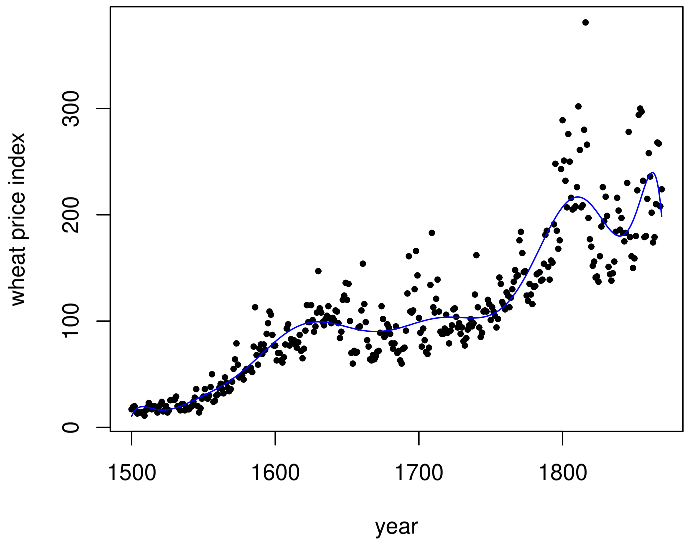
Beveridge Wheat Price Index from 1500 to 1869

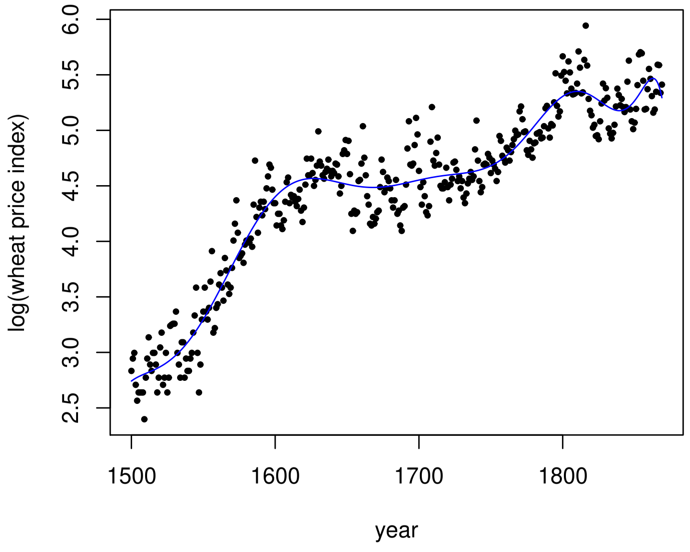


Log of Wheat Price Index

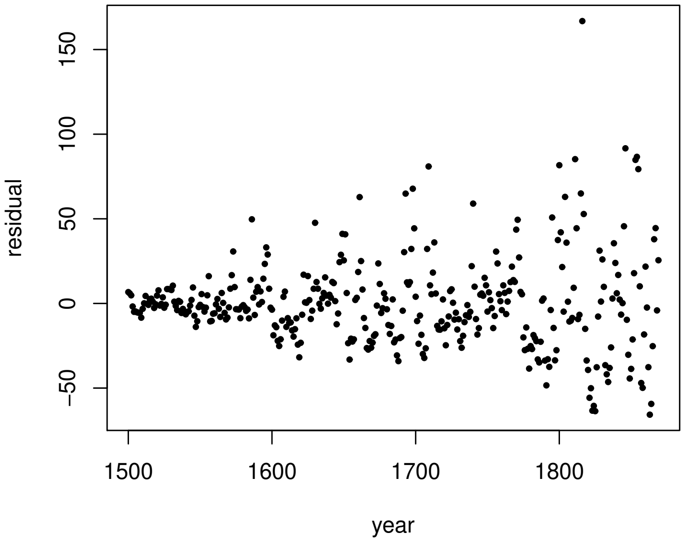


Polynomial Fits

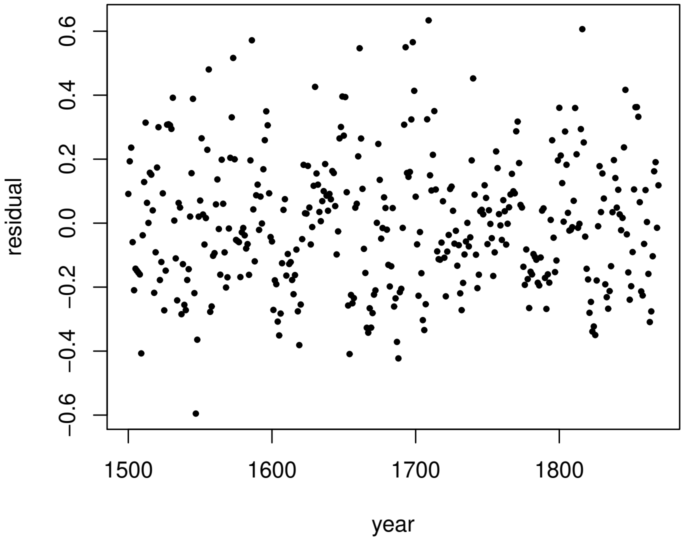




Residuals from Raw-Data Fit



Residuals from Logged-Data Fit


residual
year