Fitting Mathematical Models to Biological Data using Non-Linear Least-Squares (NLLS)

Samraat Pawar

Department of Life Sciences (Silwood Park)

Imperial College London

November 9, 2022

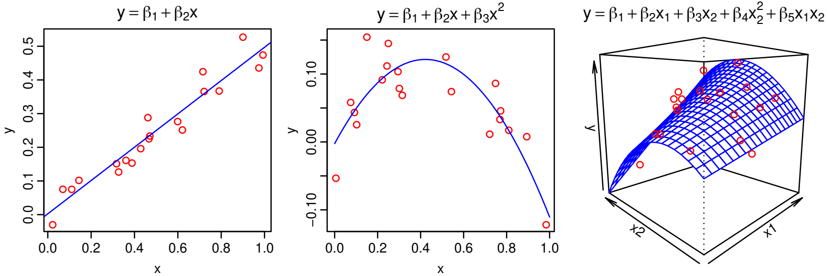
Outline

- Why NLLS?
- The NLLS fitting method
- Practicals (in `R`) overview

WHY NLLS?

LINEAR MODELS

- These are *all* good *Linear Models* (really?!):



- The data can be modelled (aka "a mathematical model fitted to them") as a *linear combination* of *variables* and *coefficients*
- Easily fitted using *Ordinary Least Squares* (OLS)
- Linear models can *include curved responses* (e.g. Polynomial regression)

What makes a model non-linear?

- OLS can be used to fit (model) equations that are *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial (quadratic): $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
  - Another quadratic: $y_i = e^{\beta_0} + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$

- What is *intrinsic linearity*? — the equation of the model to be fitted should be a *sum* of *linear terms*, i.e., the combination of *coefficient* (one of the β's) and *variables* (the x_i's):

- Some non-linear models:
  - $y_i = \beta_0 x_i^{\beta_1} + \varepsilon_i$
  - $y_i = \beta_0 + \beta_1 x_i^{\beta_2} + \varepsilon_i$
  - $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$
  - $y_i = \frac{\beta_0 x_i}{\beta_1 + x_i} + \varepsilon_i$

  In all of these, at least one term is non-linear (e.g., $x_i^{\beta_2}$, $e^{\beta_2 x_i}$, etc.)

The Least-Squares Solution

Recall what the Least Squares method does:

- Consider data on a response variable y, a predictor (independent) variable x, and n observations.
- Say we want to fit a model to these data: $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$ ($\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ are the model's $k+1$ parameters)
- An example of $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$ could be: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ (linear regression)
- The objective of any *least squares* method is to find estimates of values of the parameters ($\hat{\beta}_j$) that *minimize* the sum (S) of squared residuals (r_i) (AKA RSS):

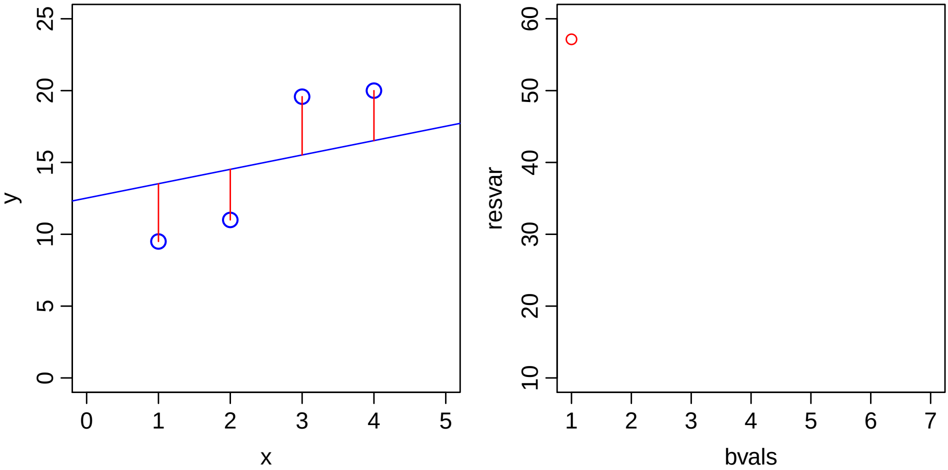
$$\mathrm{RSS} = S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$$

The Least-Squares Solution

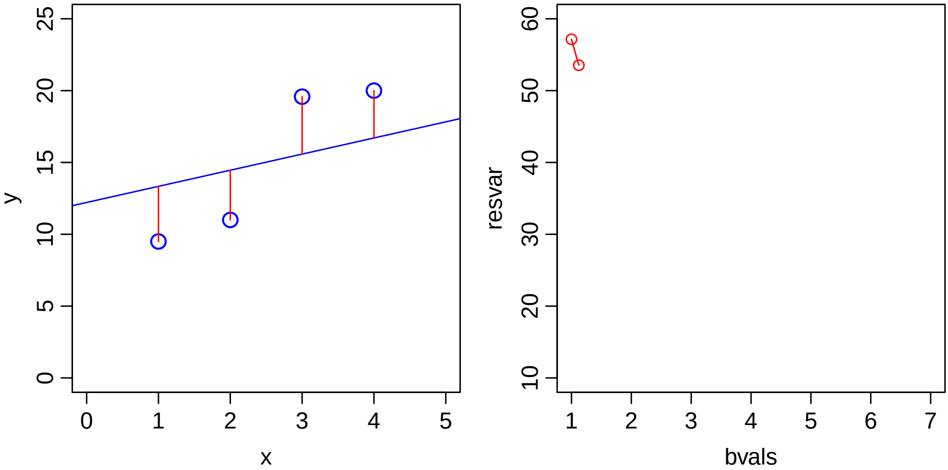
- The objective of any *least squares* method is to find estimates of values of the parameters ($\hat{\beta}_j$) that minimize the sum (S) of squared residuals (r_i) (AKA RSS):

$$\text{RSS} = S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$$

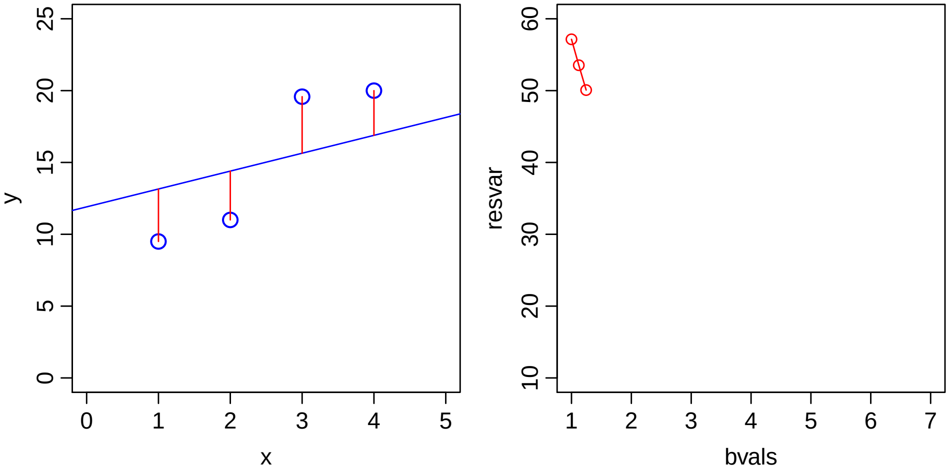
- Let's picture this using a simple (OLS) example; fitting the model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$...



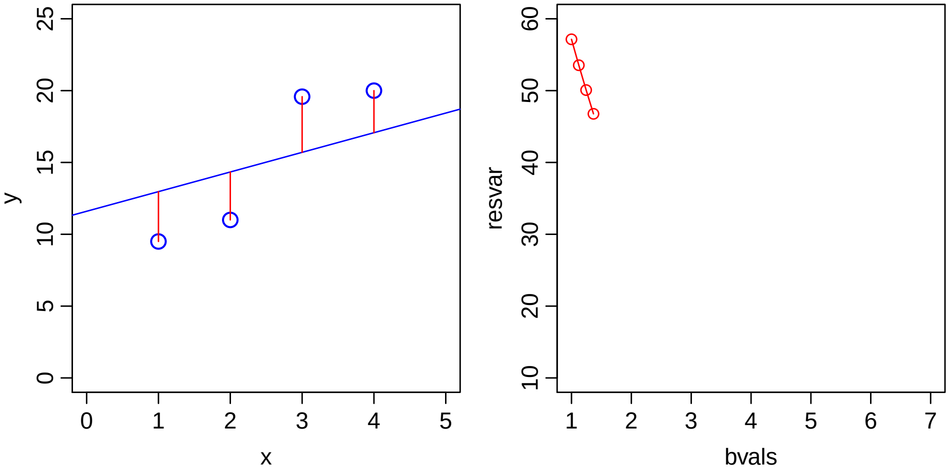
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



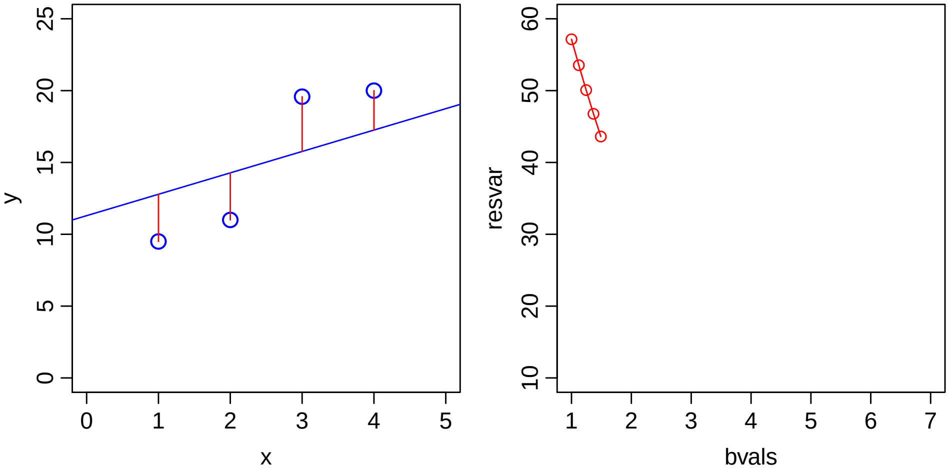
y
x
0
1
2
3
4
5
0
5
10
15
20
25
resvar
bvals
1
2
3
4
5
6
7
10
20
30
40
50
60



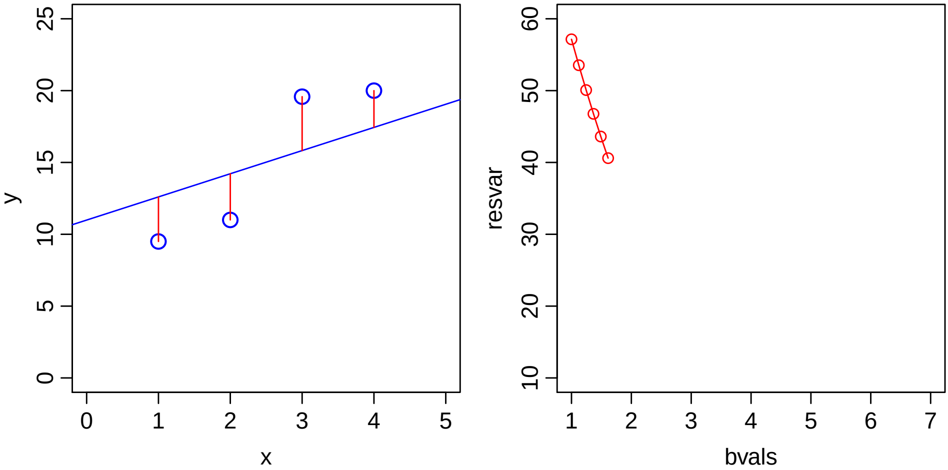
y
x
resvar
bvals



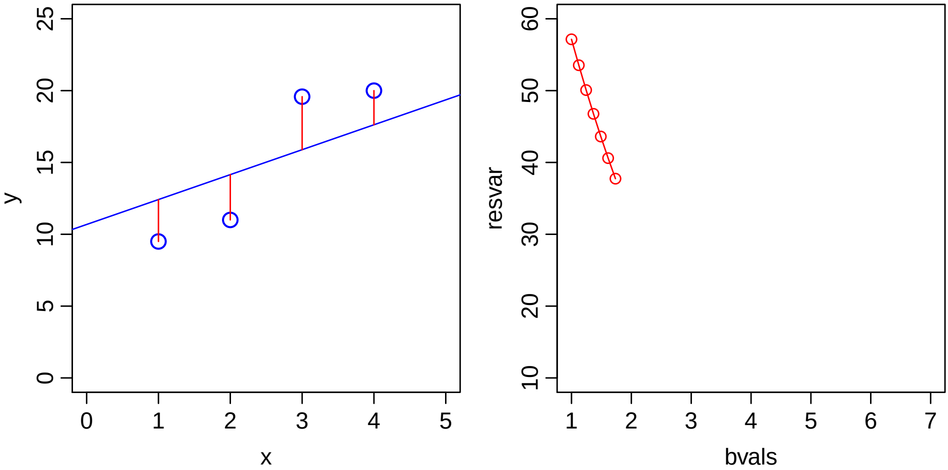
y
25
20
15
10
5
0
0
1
2
3
4
5
x
resvar
60
50
40
30
20
10
1
2
3
4
5
6
7
bvals



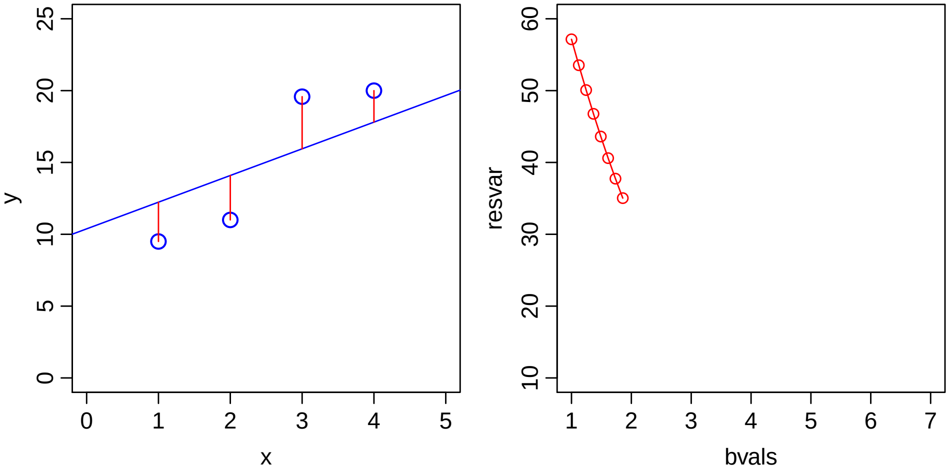
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



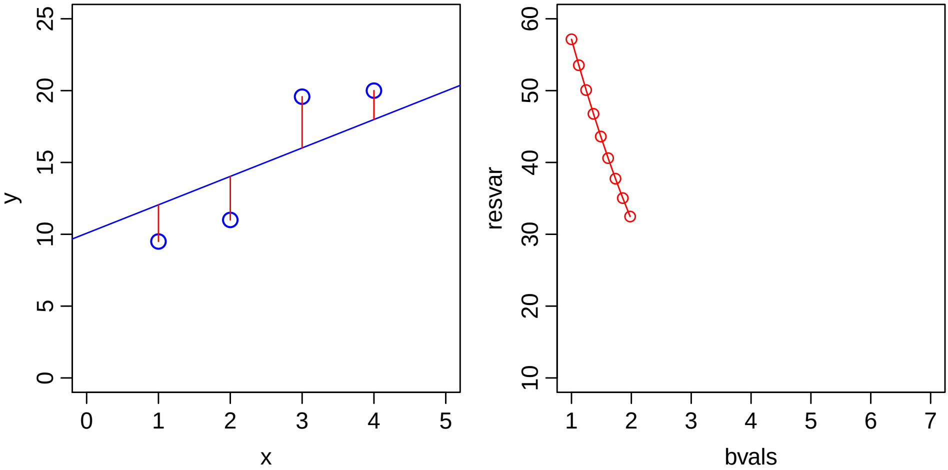
y
x
resvar
bvals



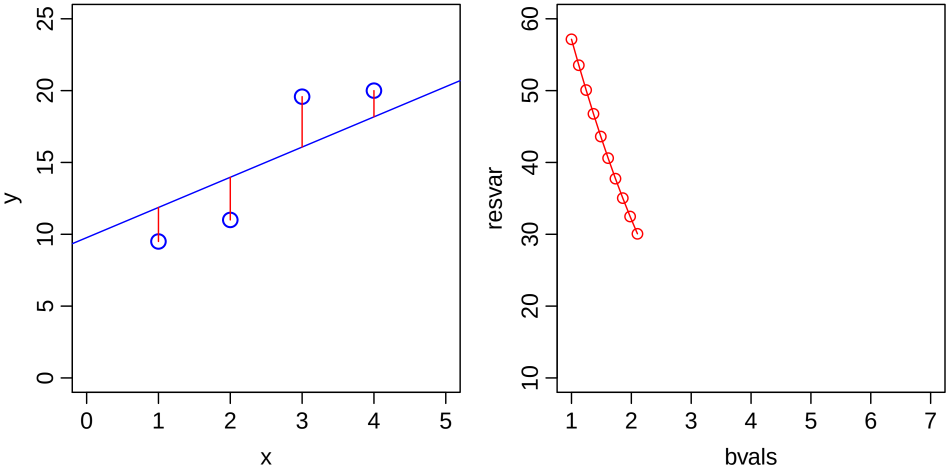
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

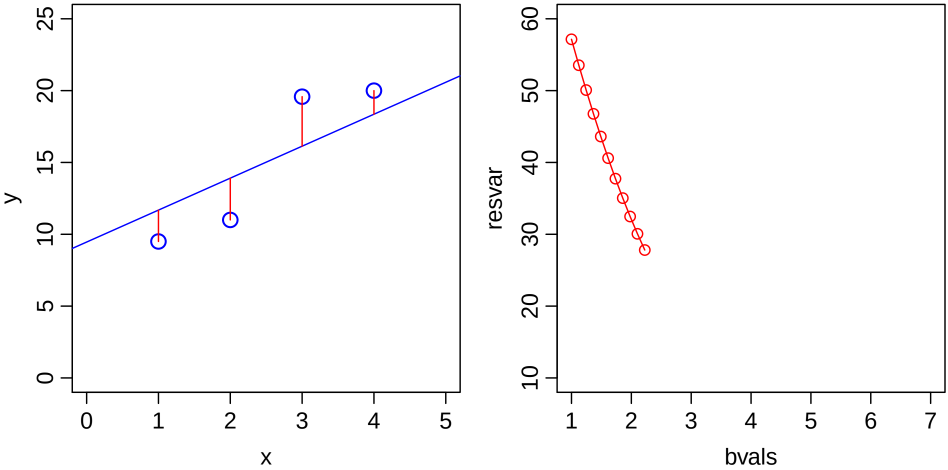


y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

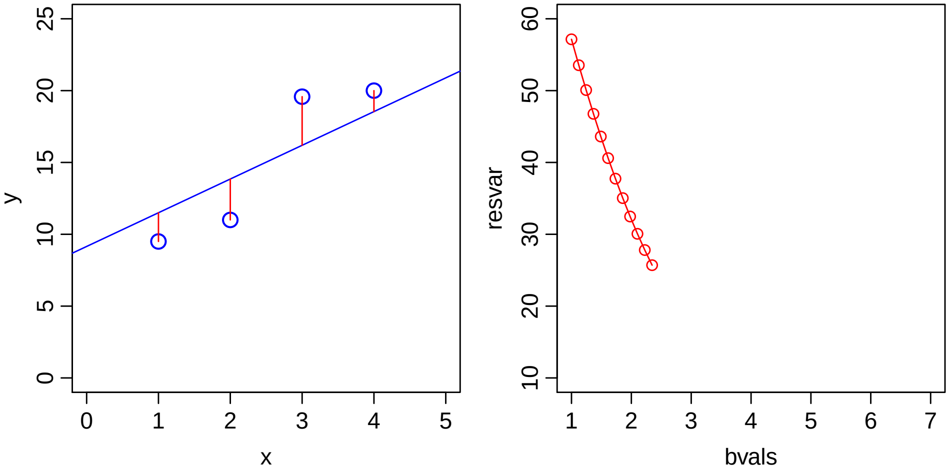


y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

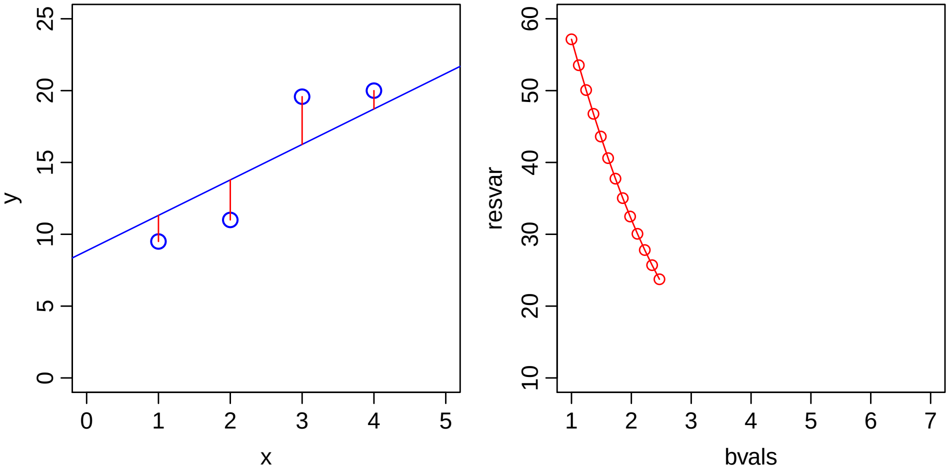

y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



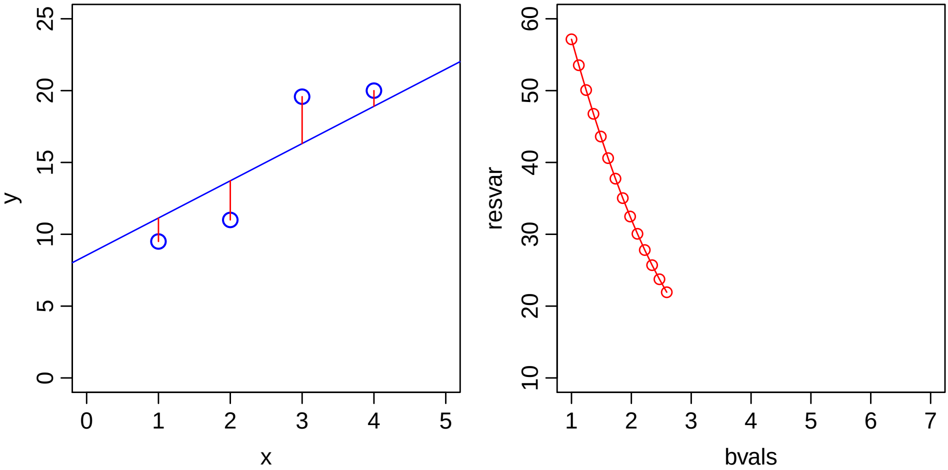
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



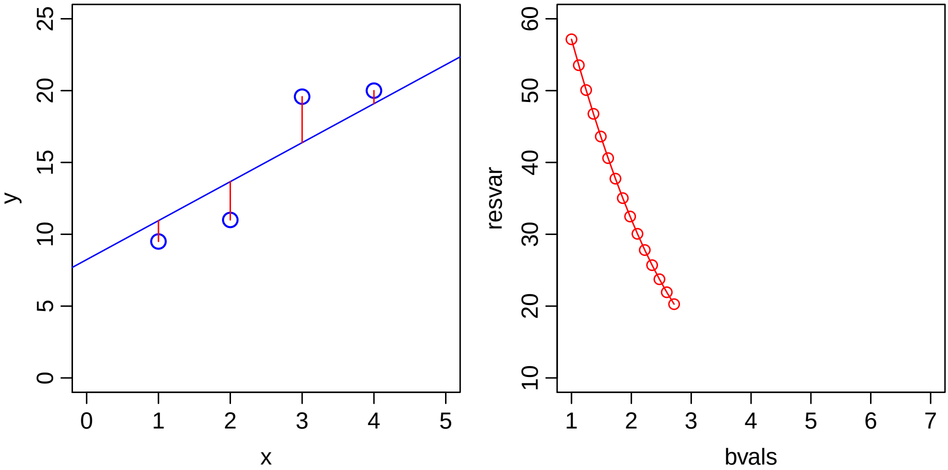
y
x
0
1
2
3
4
5
0
5
10
15
20
25
resvar
bvals
1
2
3
4
5
6
7
10
20
30
40
50
60



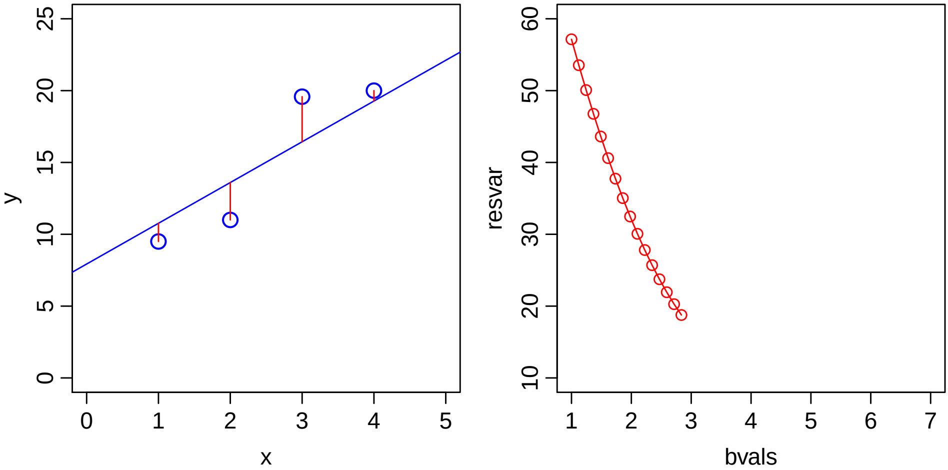
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



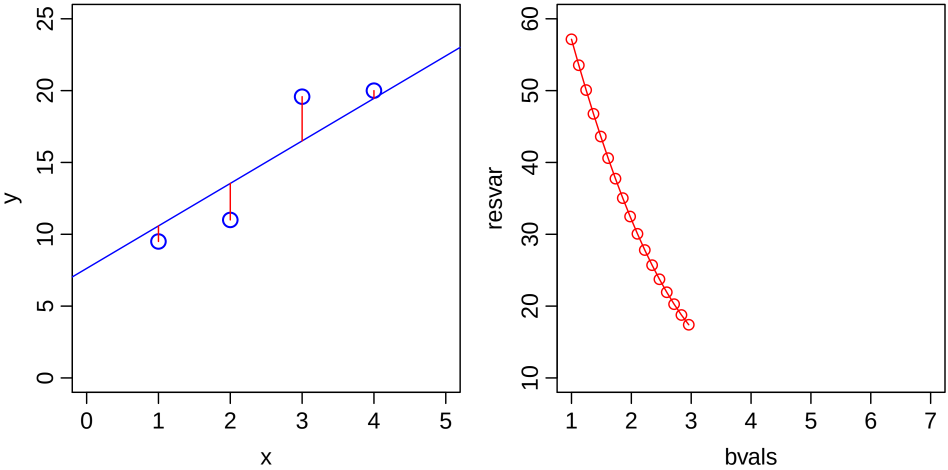
y
x
0
1
2
3
4
5
0
5
10
15
20
25
resvar
bvals
1
2
3
4
5
6
7
10
20
30
40
50
60



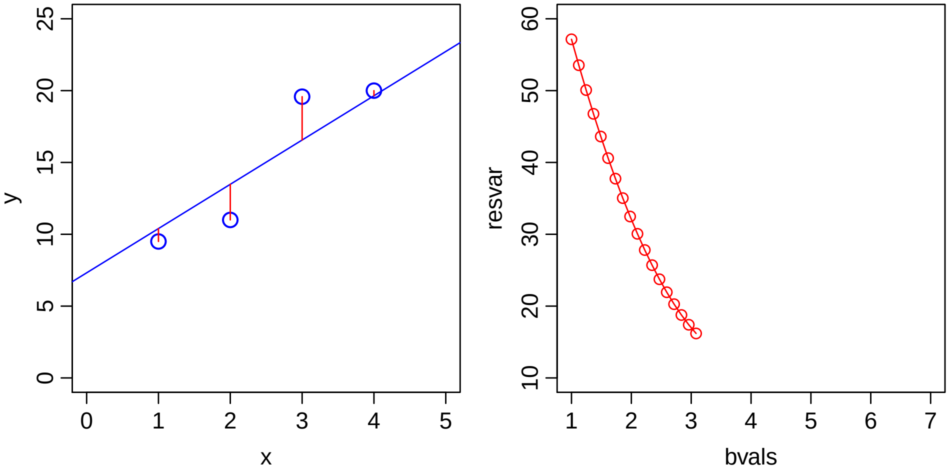
y
x
resvar
bvals
0
5
10
15
20
25
1
2
3
4
5
6
7
30
40
50
60



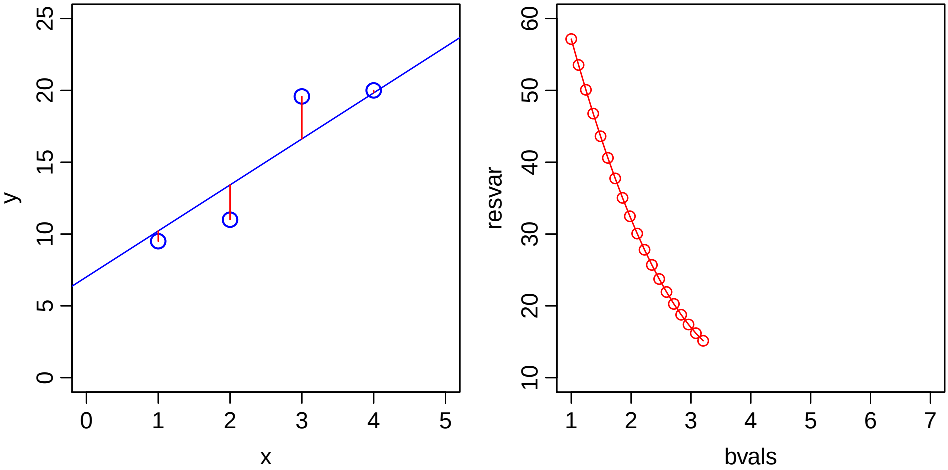
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



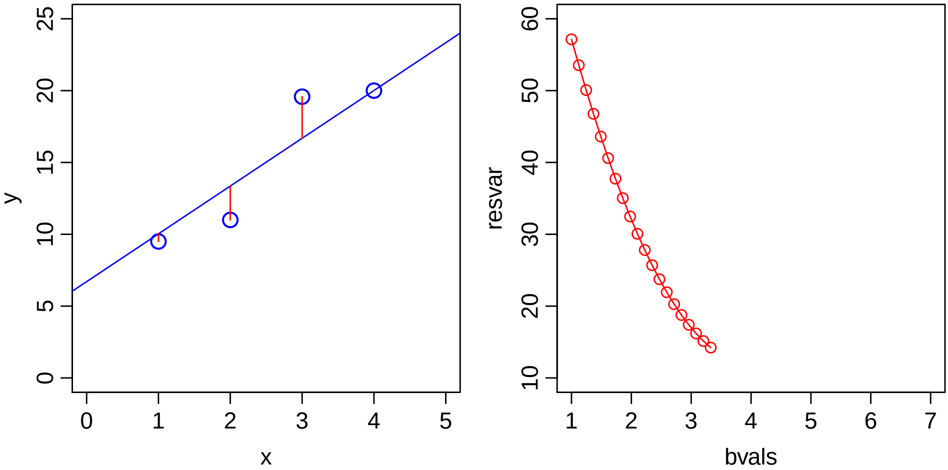
y
x
resvar
bvals



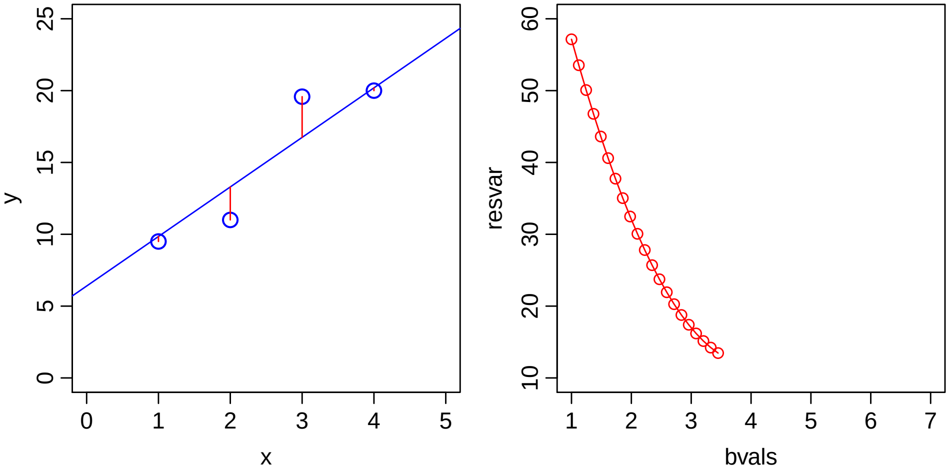
y
0
5
10
15
20
25
x
0
1
2
3
4
5
resvar
10
20
30
40
50
60
bvals
1
2
3
4
5
6
7



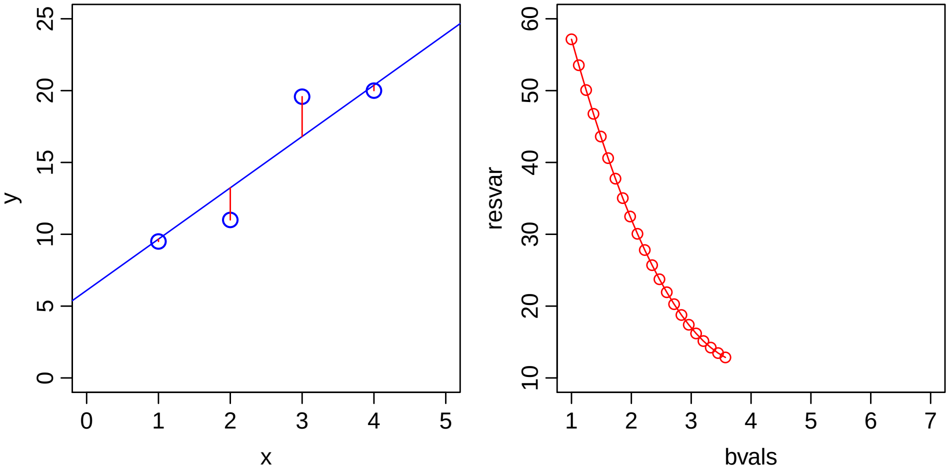
y
x
resvar
bvals
0
5
10
15
20
25
0
1
2
3
4
5
10
20
30
40
50
60
1
2
3
4
5
6
7



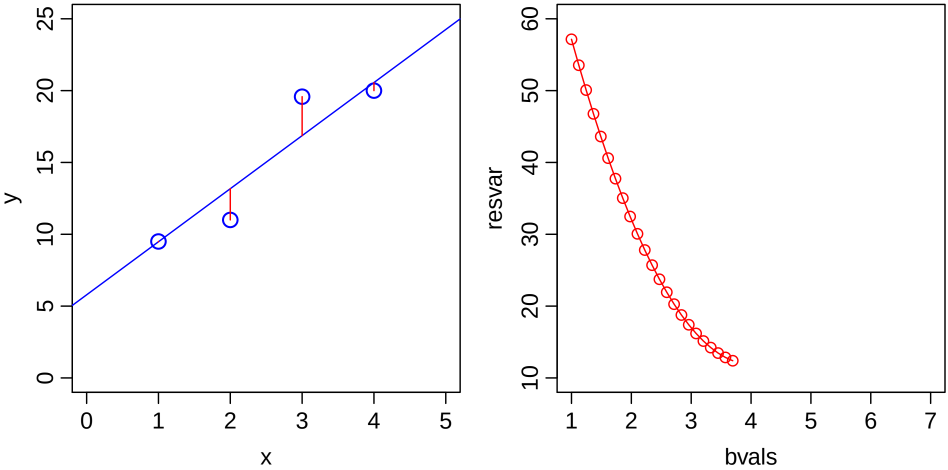
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



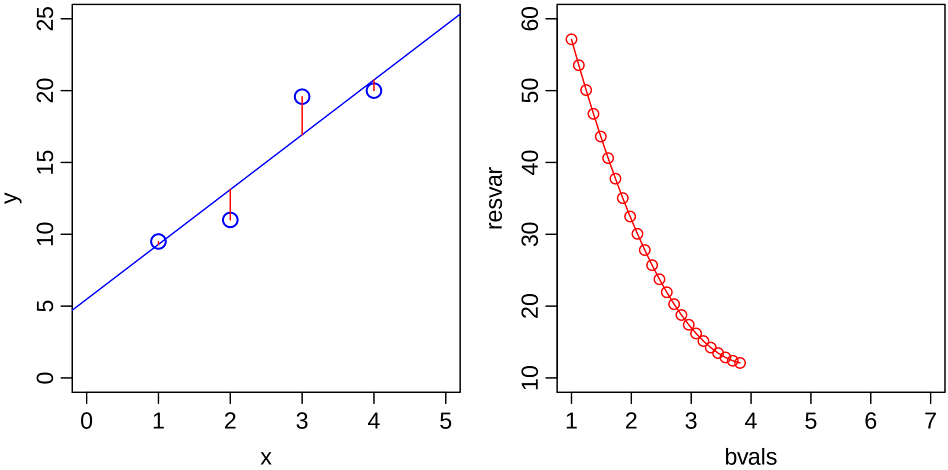
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



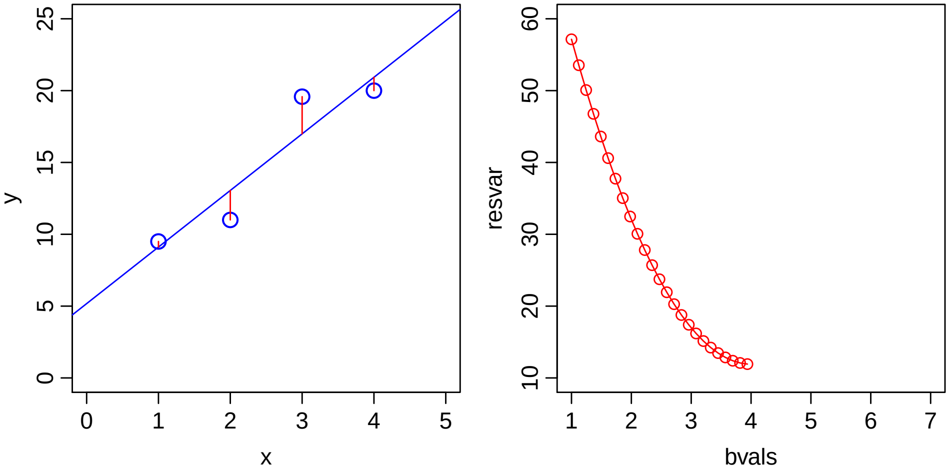
y
x
resvar
bvals



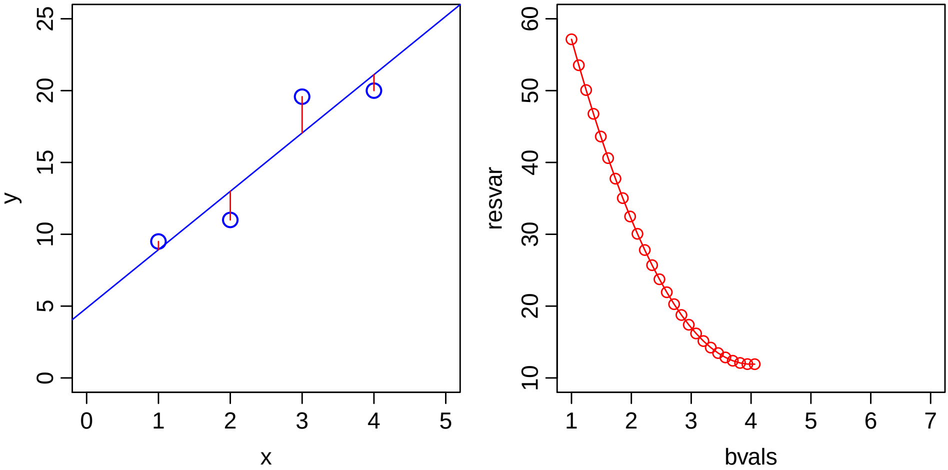
y
x
resvar
bvals



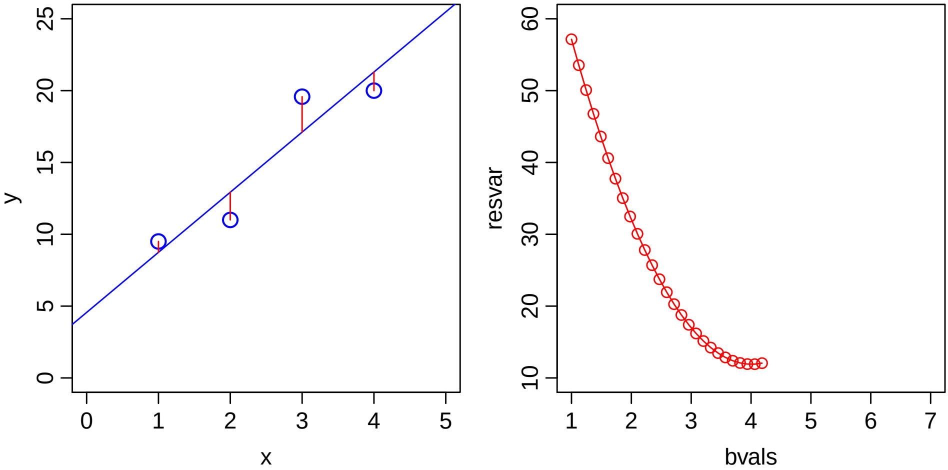
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



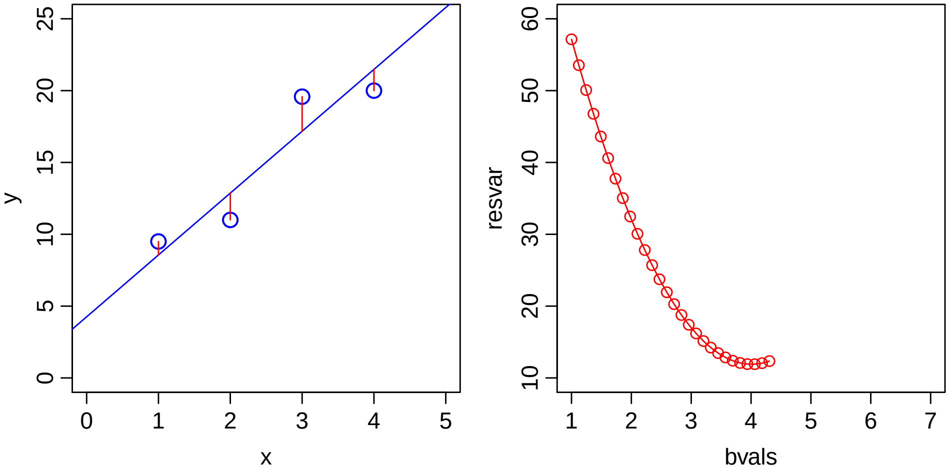
y
x
resvar
bvals



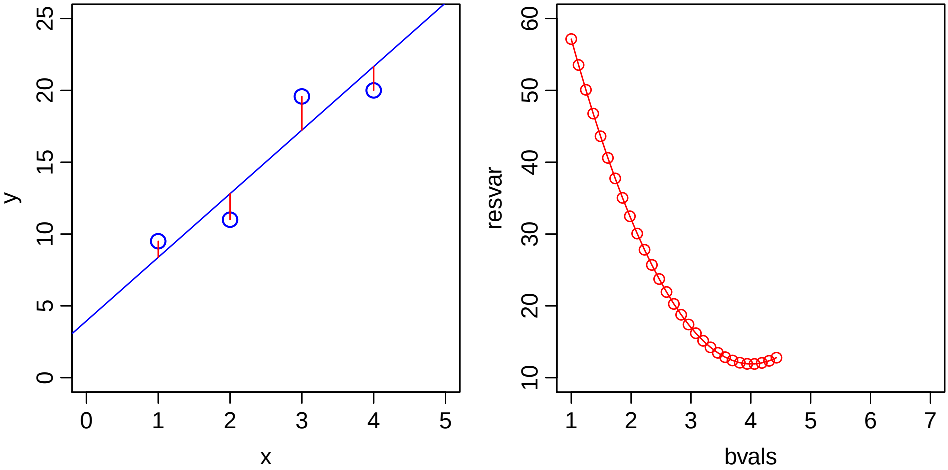
y
x
resvar
bvals



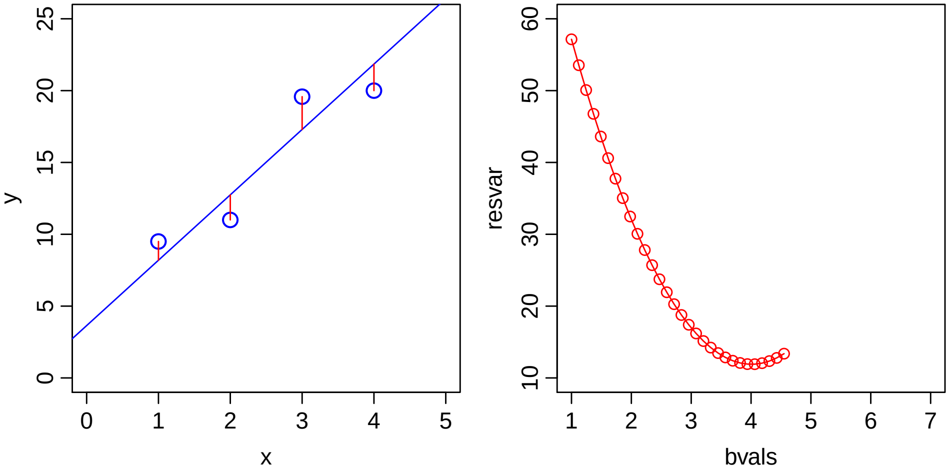
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



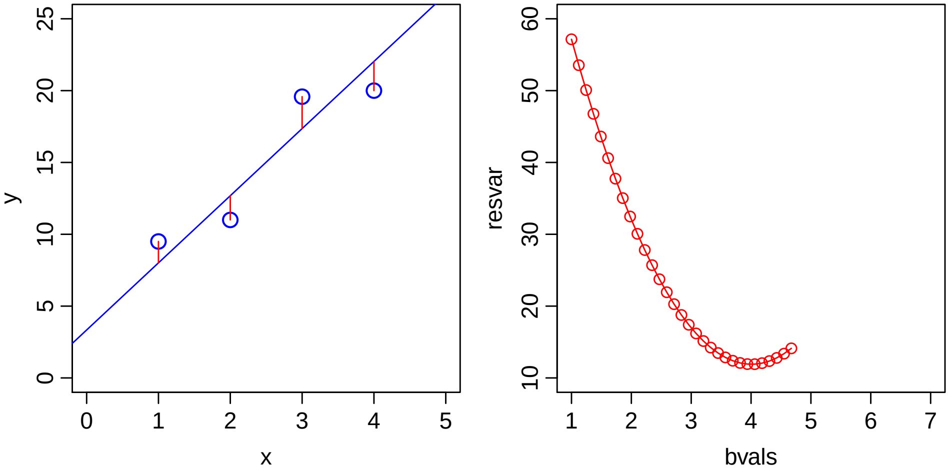
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

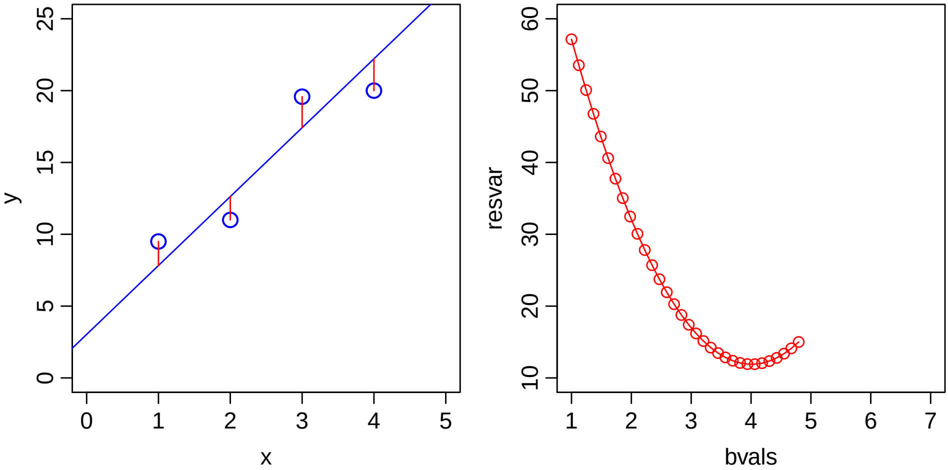


y
0
5
10
15
20
25
0
1
2
3
4
5
x
resvar
10
20
30
40
50
60
1
2
3
4
5
6
7
bvals

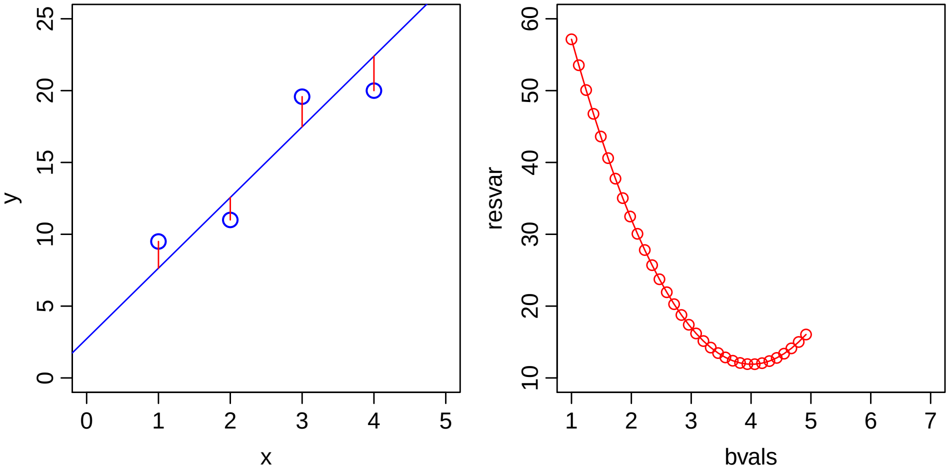


y
0
5
10
15
20
25
0
1
2
3
4
5
x
resvar
10
20
30
40
50
60
1
2
3
4
5
6
7
bvals

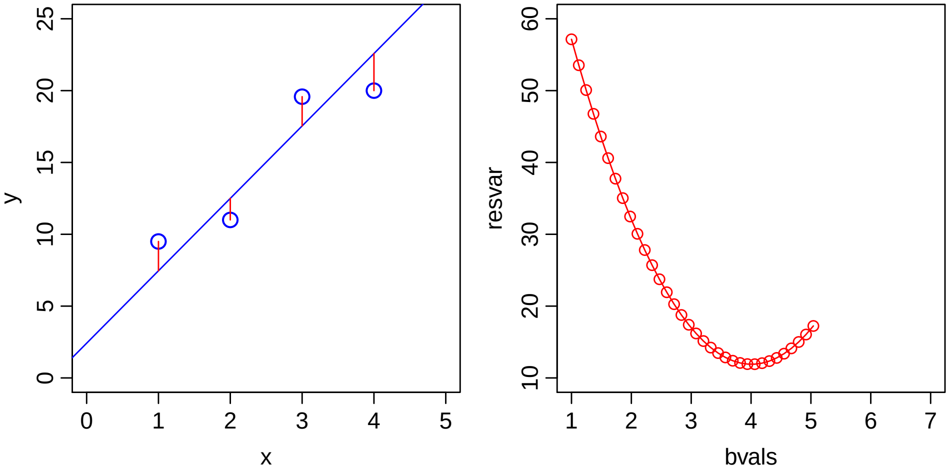

0
5
10
15
20
25
y
0
1
2
3
4
5
x
10
20
30
40
50
60
resvar
1
2
3
4
5
6
7
bvals



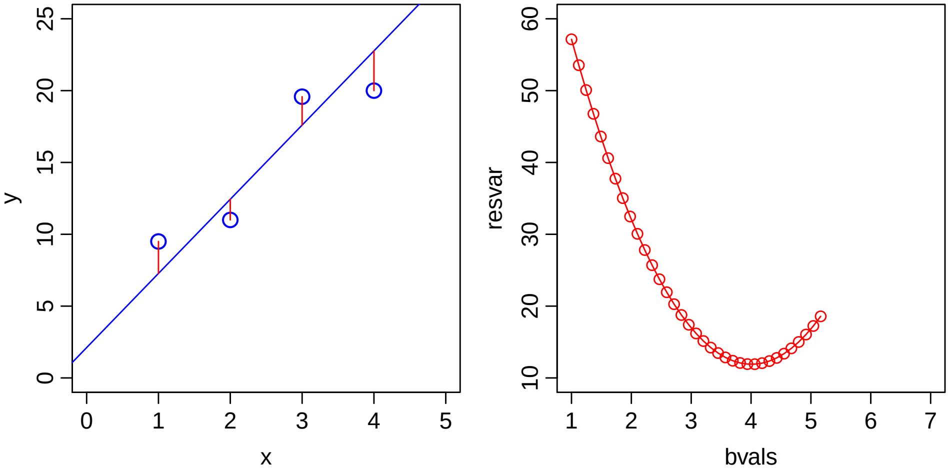
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



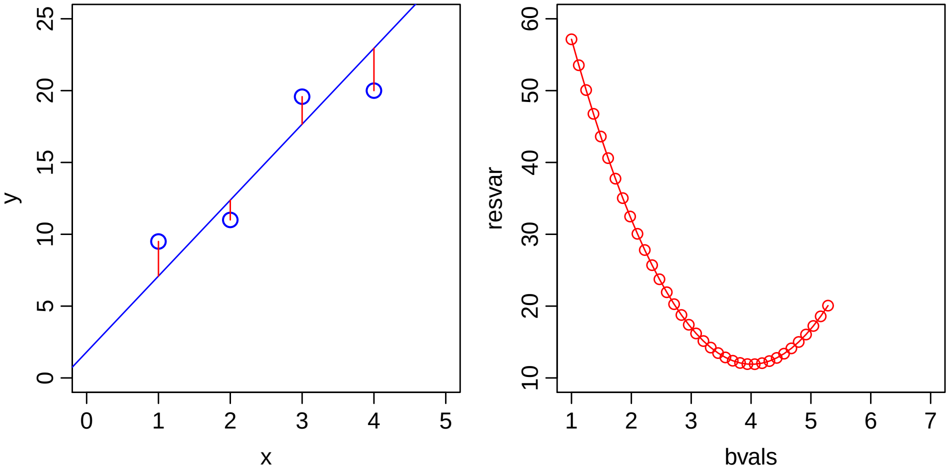
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



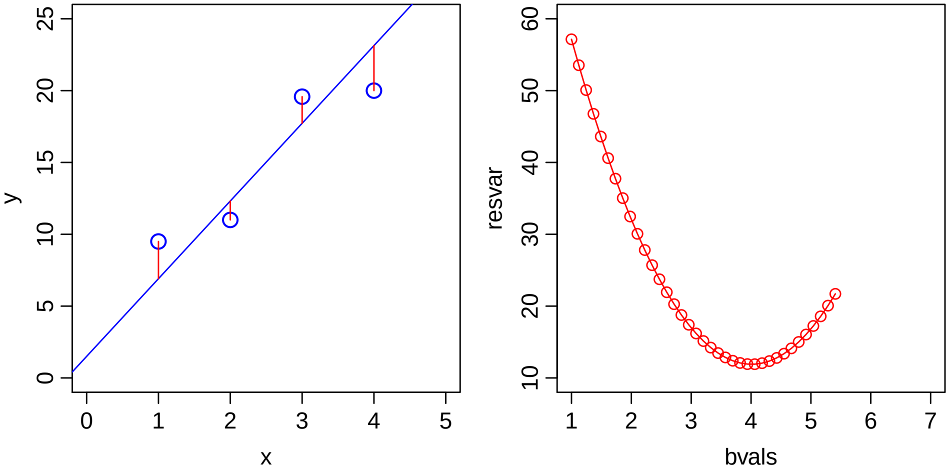
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



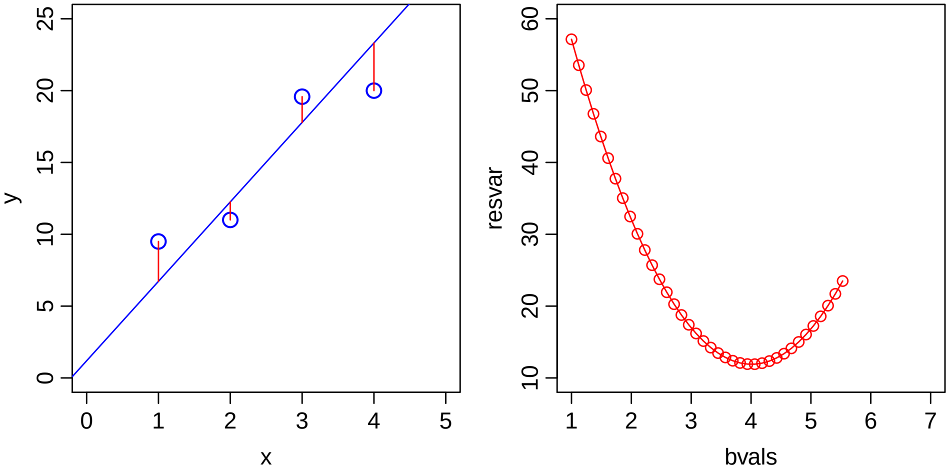
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



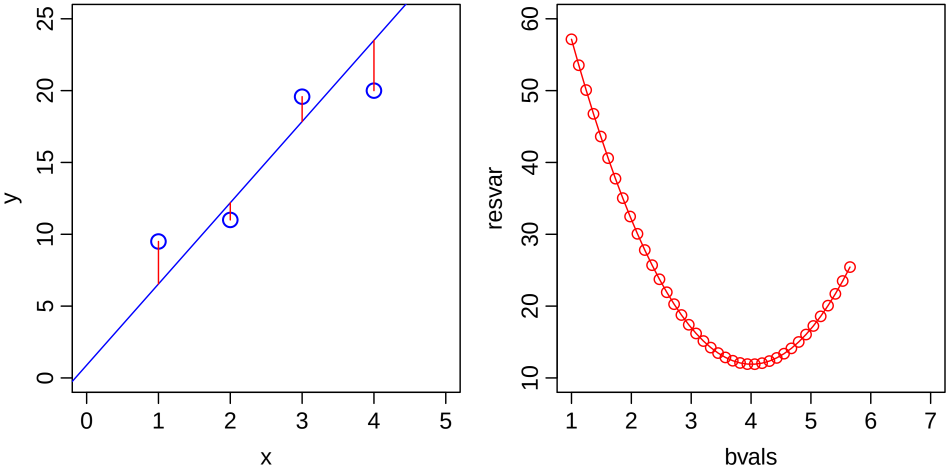
y
0
5
10
15
20
25
0
1
2
3
4
5
x
resvar
10
20
30
40
50
60
1
2
3
4
5
6
7
bvals

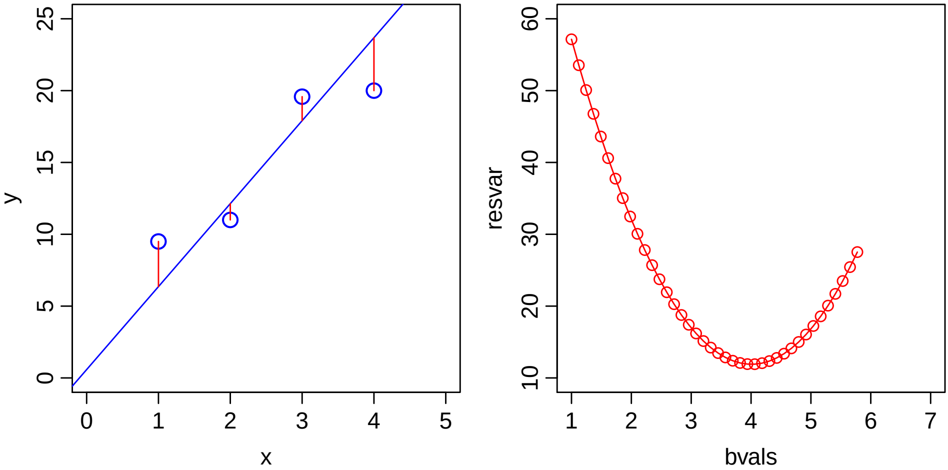


y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

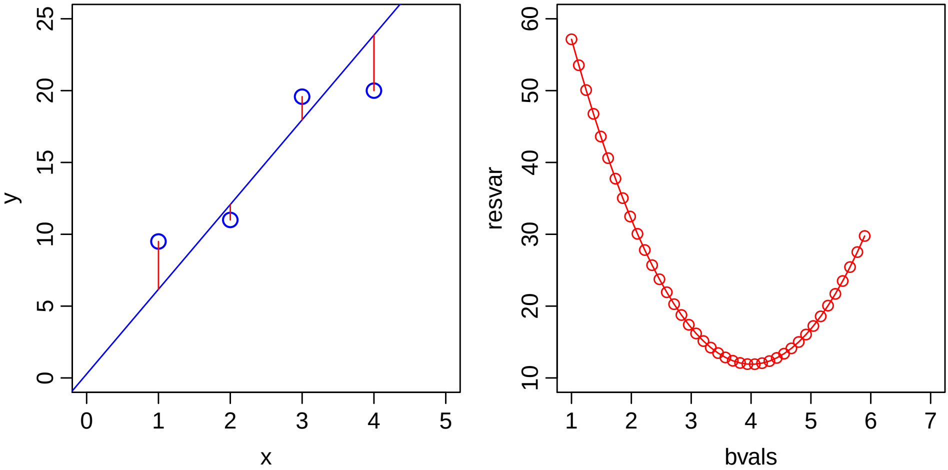


y
0
5
10
15
20
25
x
0
1
2
3
4
5
resvar
10
20
30
40
50
60
bvals
1
2
3
4
5
6
7

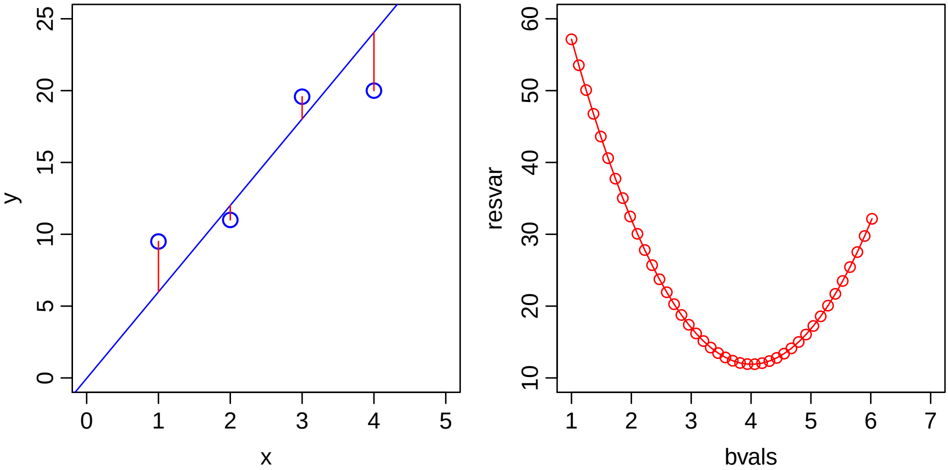

y
0
5
10
15
20
25
x
0
1
2
3
4
5
resvar
10
20
30
40
50
60
bvals
1
2
3
4
5
6
7



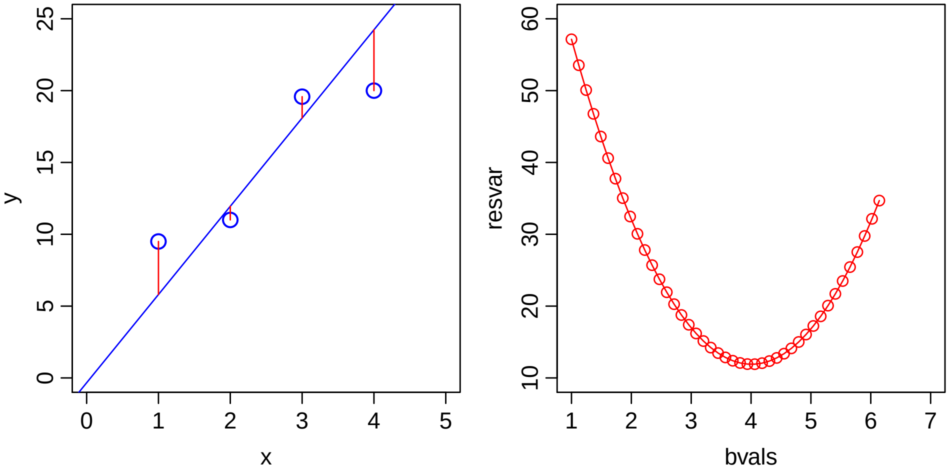
y
0
5
10
15
20
25
x
0
1
2
3
4
5
resvar
10
20
30
40
50
60
bvals
1
2
3
4
5
6
7



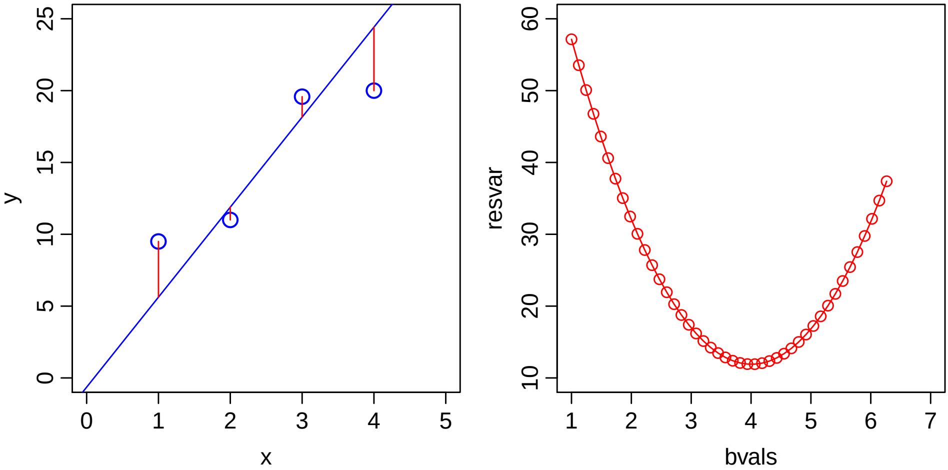
y
x
resvar
bvals



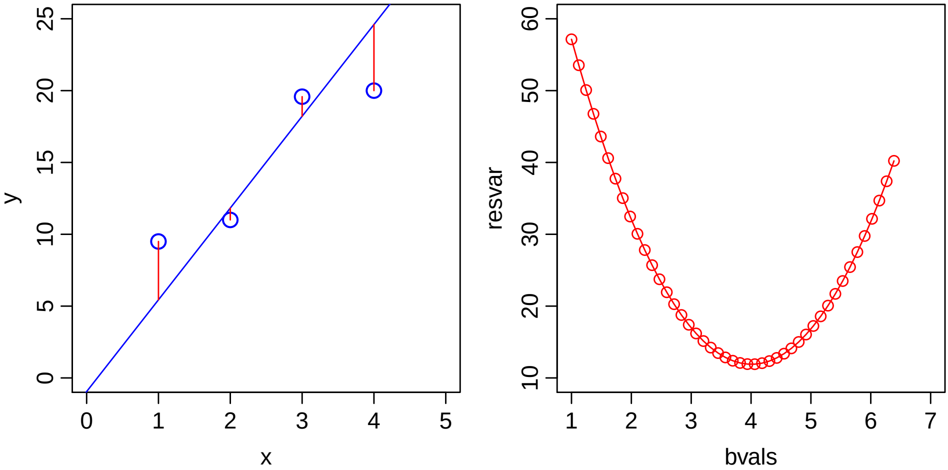
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



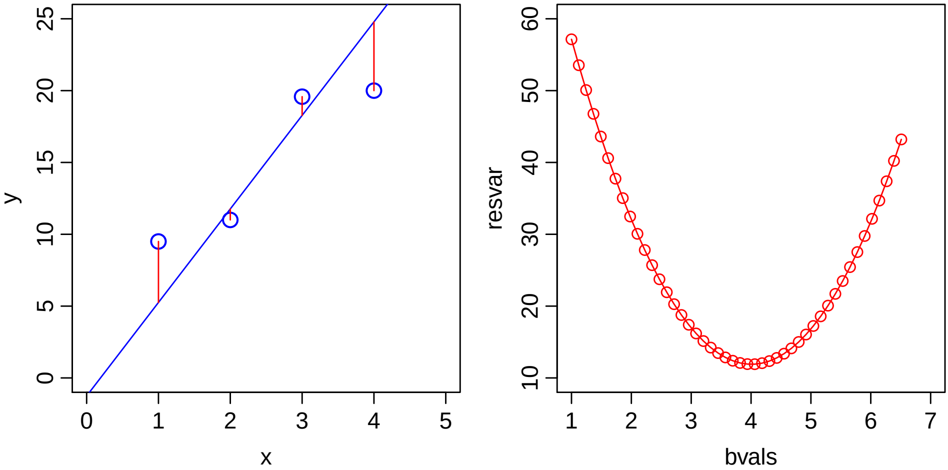
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



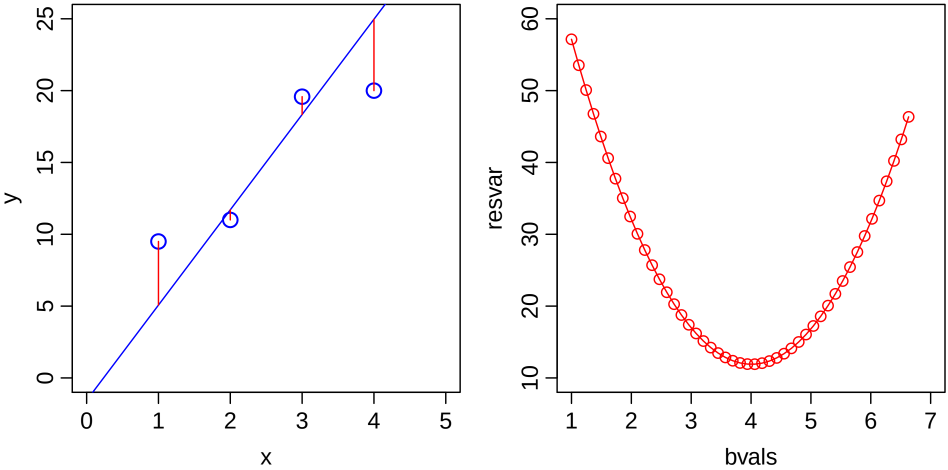
y
0
5
10
15
20
25
x
0
1
2
3
4
5
resvar
10
20
30
40
50
60
bvals
1
2
3
4
5
6
7



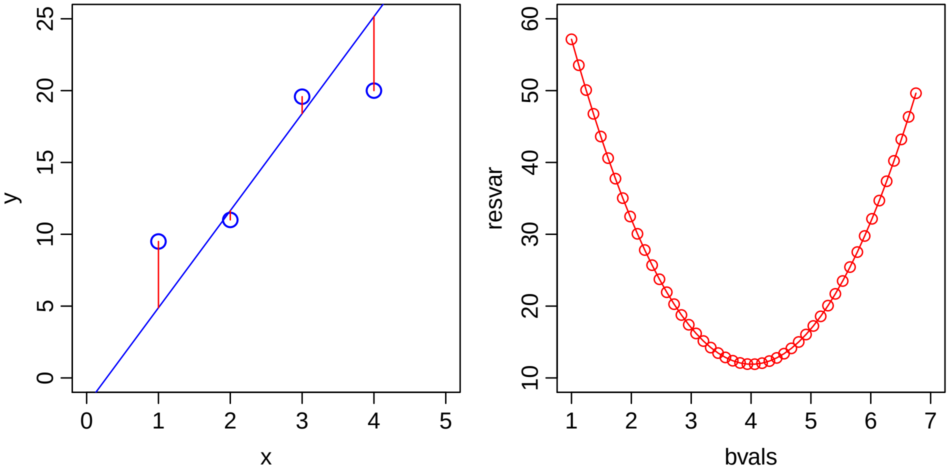
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



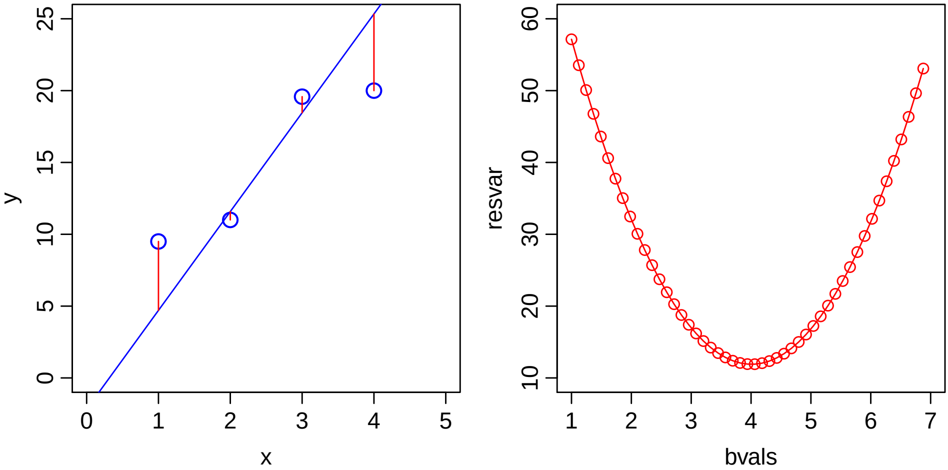
y
x
0
1
2
3
4
5
10
15
20
25
resvar
bvals
6
7
30
40
50
60



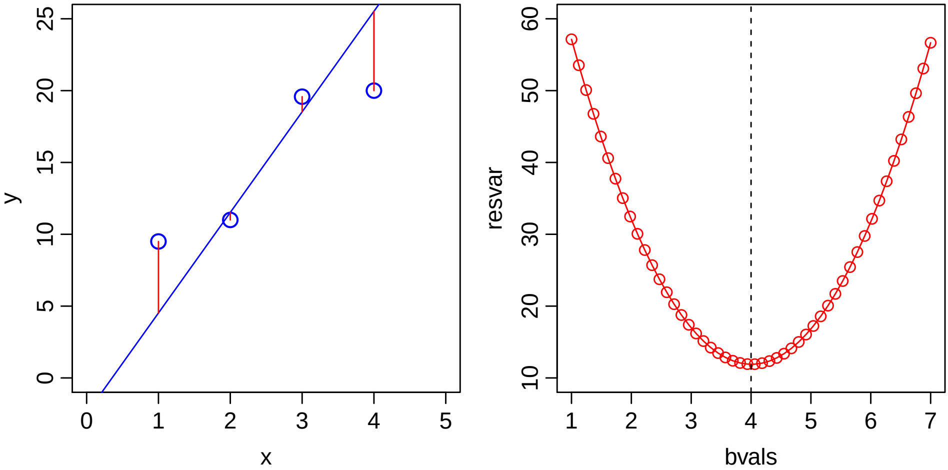
y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7



0
5
10
15
20
25
y
0
1
2
3
4
5
x
10
20
30
40
50
60
resvar
1
2
3
4
5
6
7
bvals

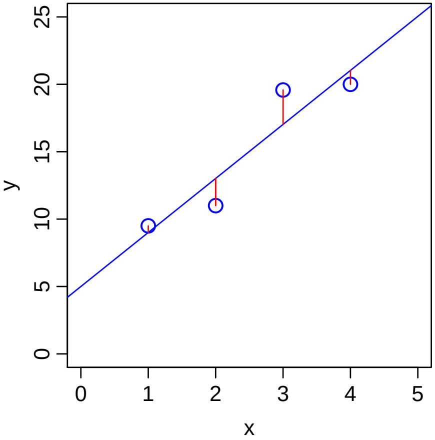


0
5
10
15
20
25
y
0
1
2
3
4
5
x
10
20
30
40
50
60
resvar
1
2
3
4
5
6
7
bvals



y
x
0
5
10
15
20
25
0
1
2
3
4
5
resvar
bvals
10
20
30
40
50
60
1
2
3
4
5
6
7

If the model is linear, the Least-Square solution is exact



$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$9.50 = 5 + 4 \times 1 + 0.50$$
$$11.00 = 5 + 4 \times 2 - 2.00$$
$$19.58 = 5 + 4 \times 3 + 2.58$$
$$20.00 = 5 + 4 \times 4 - 1.00$$

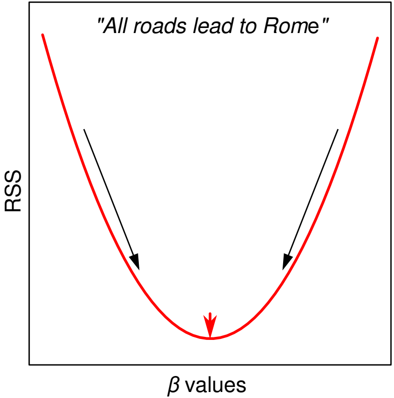
The least squares solution here is: $\beta_0 = 5; \beta_1 = 4$

- *This system of (linear) equations can be compactly represented (and solved using matrix algebra) as* $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

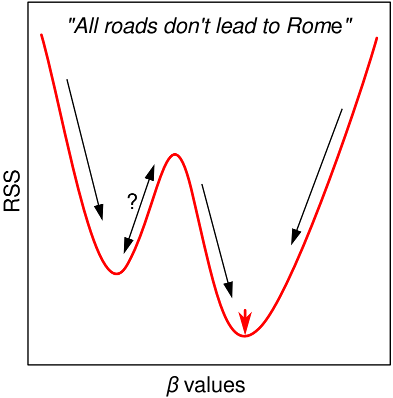
INTRINSIC NON-LINEARITY MAKES LEAST-AQUARES MODEL FITTING DIFFICULT

- In an intrinsically non-linear model such as $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$, the nice trick of solving $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ *exactly* is impossible

Linear Least-Squares Minimization



Non-Linear Least-Squares Minimization



OK, FINE, WHY WOULD *I* EVER NEED NLLS?

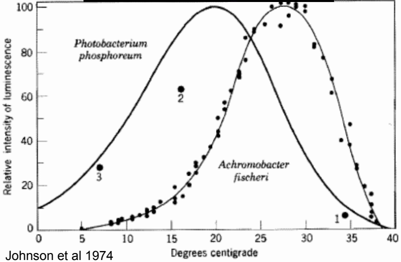
- Many observations in biology are just *not* well-fitted by a linear model

- That is, the underlying biological phenomena/phenomenon are not well-described by a linear equation

- Examples:
  - Michaelis-Menten biochemical (reaction) kinetics
  - Allometric growth
  - Responses of metabolic rates to changing temperature
  - Consumer-Resource (e.g., predator-prey) functional responses
  - Individual growth
  - Population growth
  - Time-series data (e.g., fitting a sinusoidal function)

- *Can you think of some examples?*

NON-LINEAR MODEL EXAMPLE: TEMPERATURE AND METABOLISM



Johnson et al 1974

$$B = B_0 \boxed{e^{-\frac{E}{kT}}} f(T, T_{pk}, E_D)$$

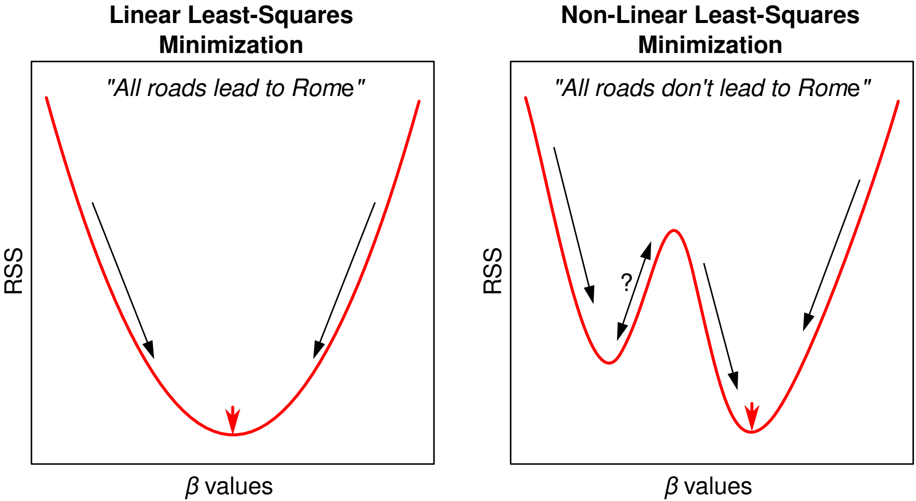
T = temperature (K)
k = Boltzmann constant (eV K^{-1})
E = Activation energy (eV)
T_{pk} = Temperature of peak performance
E_D = Deactivation energy (eV)
(J H van't Hoff 1884, S Arrhenius 1889)

THE NLLS FITTING METHOD

THE NLLS METHOD: OVERVIEW

- OK, so we cannot find an exact, simple solution to the least-squares problem for non-linear models
- But we can use a computer to find a *approximate but close-to-optimal* least-squares solution as follows:
  - Choose starting (initial values for the parameters we want to estimate (β_j's)
  - Then, adjust the parameters *iteratively* (using a specific "algorithm" that is better than searching *randomly*) such that the RSS is gradually decreased
  - Eventually, if it all goes well, a combination of β_j's that is *very close* to the desired solution (where the RSS is *approximately* minimized) can be found

THE NLLS FITTING / OPTIMIZATION PROCESS



The NLLS fitting / optimization process

The general procedure / algorithm is:

1. Start with an initial value for each parameter in the model
2. Generate the curve defined by the initial values
3. Calculate the residual sum-of-squares (RSS)
4. Adjust the parameters to make the curve come closer to the data points. *This the tricky part — more on this in the next slide*
5. Adjust the parameters again so that the curve comes even closer to the points (RSS decreases)
6. Repeat 4–5
7. Stop simulations when the adjustments make virtually no difference to the RSS

NLLS fitting / optimization algorithms

The *tricky part — adjust parameters to make curve come closer to the data points* (step 4) — has two main algorithms that are generally used:

- The **Gauss-Newton** algorithm is often used, but doesn't work very well if the model to be fitted is mathematically complicated (the parameter search "landscape" is difficult) and the *starting values* for parameters are far-off-optimal
- The **Levenberg-Marquardt** algorithm switches between Gauss-Newton and "gradient descent" and is more robust against starting values that are far-off-optimal and is more reliable in most scenarios.

NLLS FITS – ASSESSMENT AND REPORTING

- Once the NLLS fitting is done, you need to get the *goodness of fit measures*
- First, of course, examine the fits visually
- Report the goodness-fit results:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - Estimated coefficients
  - For each coefficient, standard error (can be used for CI's), t-statistic and corresponding (two-tailed) p-value
- You will learn to calculate all these in the practicals
- You may also want to *compare and select between multiple competing models*
- Unlike in Linear Models, R^2 values *should not* be used to interpret the quality of a NLLS fit (more on this in the practicals).

NLLS ASSUMPTIONS

NLLS-regression has all the assumptions of OLS-regression:

- No (in practice, minimal) measurement error in explanatory variable (x-axis variable)
- Data have constant normal variance — errors in the y-axis are homogeneously distributed over the x-axis range
- The measurement/observation errors are Normally distributed (Gaussian)
- What if the errors are not normal? — Interpret results cautiously, and use Maximum Likelihood or Bayesian fitting methods instead

PRACTICALS OVERVIEW

NLLS Fitting Practicals

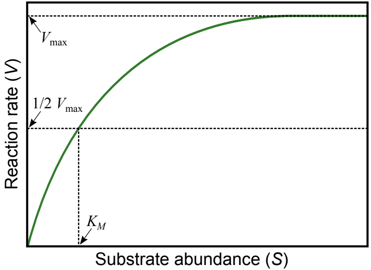
- We will use `R`
- For fitting simple non-linear models, the `nls` function in R is sufficient
  - It uses the **Gauss-Newton** algorithm by default
  - The command is `nls()`
  - It is part of the `stats` base package (so no extra installation and loading of package necessary)
- For fitting complex non-linear models the **Levenberg-Marquardt (LM)** algorithm is better
  - The command is `nlsLM()`
  - It is available through the the `minpack.lm` package `http://cran.r-project.org/web/packages/minpack.lm`
  - It offers additional features like the ability to “bound” parameters to realistic values

NLLS Fitting Practicals

- We will start with NLLS fitting of the Michaelis-Menten model of biochemical reaction kinetics:

$$V = \frac{V_{\max}[S]}{K_m + [S]}$$

- S = Substrate density
- $V_{\max}$ = Maximum reaction rate (at saturating substrate concentration)
- K_M = Half-saturation constant; the S at which reaction rate reaches half of possible maximum ($= \frac{1}{2} V_{\max}$)



- You will use NLLS fitting to obtain estimates of $V_{\max}$ and K_M
- Note that $V_{\max} \leq 0$ and $K_M \leq 0$ are physically impossible (useful fir picking starting values)

READINGS

- Motulsky, Harvey, and Arthur Christopoulos. Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting. OUP USA, 2004.