Fitting mathematical models to data using Non-linear Least-Squares

Samraat Pawar

Department of Life Sciences

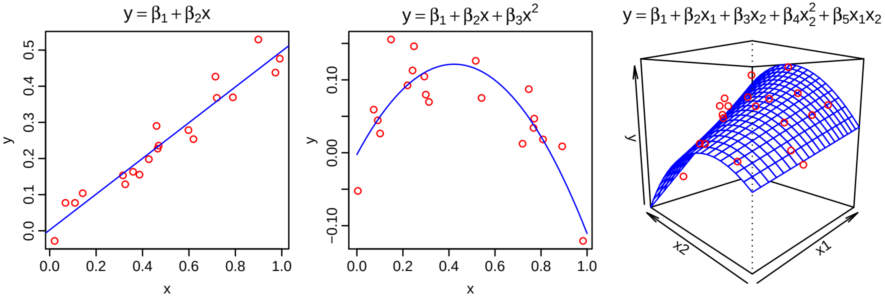
(Silwood Park)

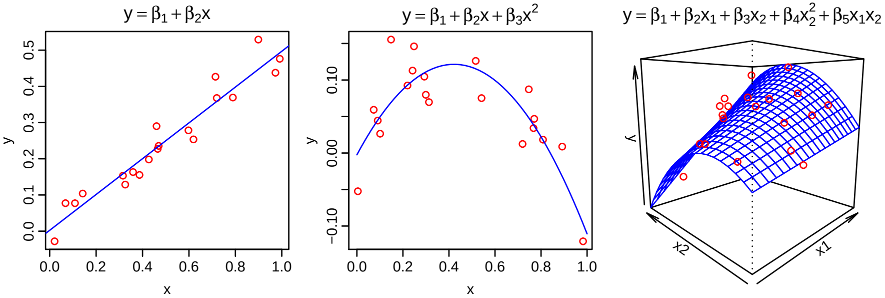
July 13, 2018

OUTLINE

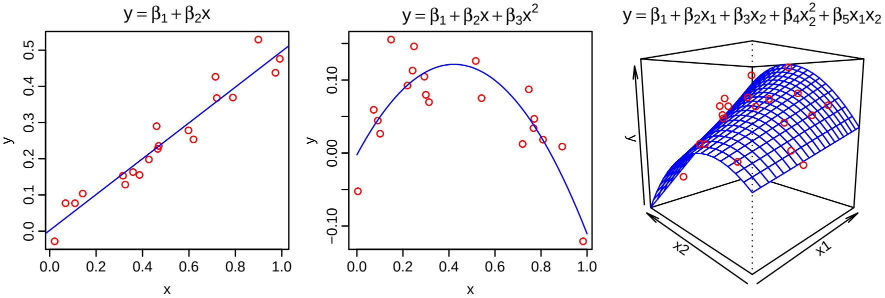
- Why Non-Linear Least Squares regression / fitting?
- The NLLS fitting method
- NLLS in `R`
- Afternoon practicals overview

LINEAR MODELS ARE GREAT



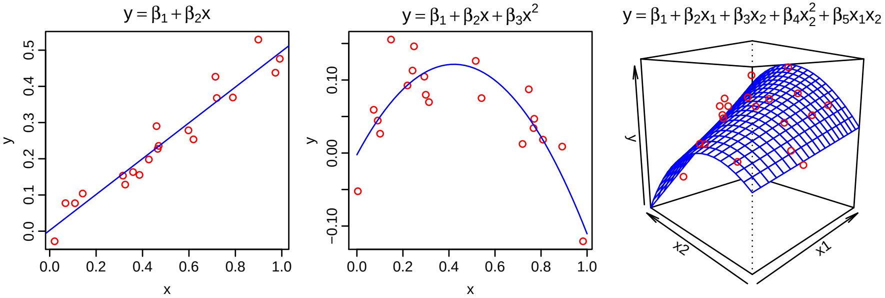


- These are *all* good *Linear Models* (really?!)



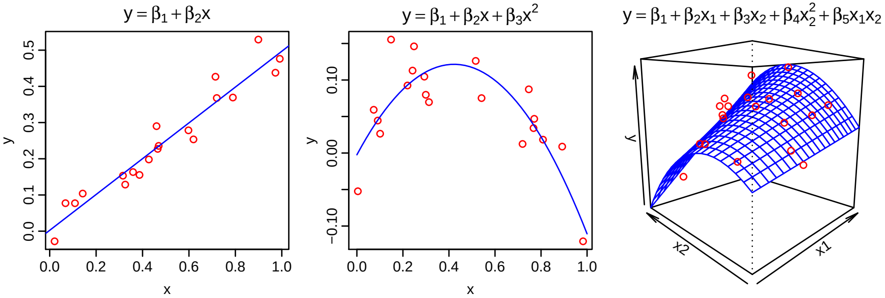
- These are *all* good *Linear Models* (really?!)
- The data can be modelled (aka "fitted to a mathematical model") as a *linear combination* of *variables* and *coefficients*

Linear models are great

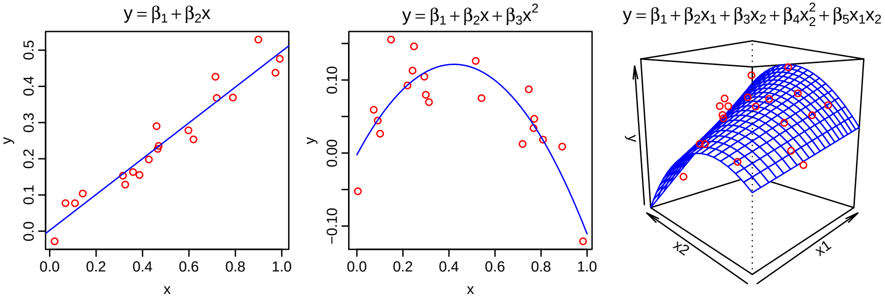


- These are *all* good *Linear Models* (really?!)
- The data can be modelled (aka "fitted to a mathematical model") as a *linear combination* of *variables* and *coefficients*
- Easily fitted using *Ordinary Least Squares* (OLS) regression

Linear models are great



- These are *all* good *Linear Models* (really?!)
- The data can be modelled (aka "fitted to a mathematical model") as a *linear combination* of *variables* and *coefficients*
- Easily fitted using *Ordinary Least Squares* (OLS) regression
- Linear models can *include curved responses* (e.g. polynomial regression)



- These are *all* good *Linear Models* (really?!)
- The data can be modelled (aka "fitted to a mathematical model") as a *linear combination* of *variables* and *coefficients*
- Easily fitted using *Ordinary Least Squares* (OLS) regression
- Linear models can *include curved responses* (e.g. polynomial regression)
- OK, so then *why Non-Linear Least Squares (NLLS) fitting?*

Why NLLS? – first, what makes a model non-linear?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$

Why NLLS? – first, what makes a model non-linear?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*
- Are these models linear in their *parameters*?

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*
- Are these models linear in their *parameters*?
  - $y_i = \beta_0 + \beta_1 x_i^{\beta_2} + \varepsilon_i$

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*
- Are these models linear in their *parameters*?
  - $y_i = \beta_0 + \beta_1 x_i^{\beta_2} + \varepsilon_i$
  - $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$

Why NLLS? – first, what makes a model non-linear?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*
- Are these models linear in their *parameters*?
  - $y_i = \beta_0 + \beta_1 x_i^{\beta_2} + \varepsilon_i$
  - $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$

WHY NLLS? – FIRST, WHAT MAKES A MODEL NON-LINEAR?

- OLS can be used to fit both linear and nonlinear *equations* that *intrinsically linear*, e.g.,
  - Straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
  - Polynomial: $y_i = \exp(\beta_0) + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
- Indeed, for OLS to work, we need *intrinsic linearity* — i.e., the equation to be fitted (model) should be *linear in the parameters*
- Are these models linear in their *parameters*?
  - $y_i = \beta_0 + \beta_1 x_i^{\beta_2} + \varepsilon_i$
  - $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$

  NO!

SO WHAT — WHY IS INTRINSIC NON-LINEARITY A PROBLEM?

Recall what the Least Squares method does:

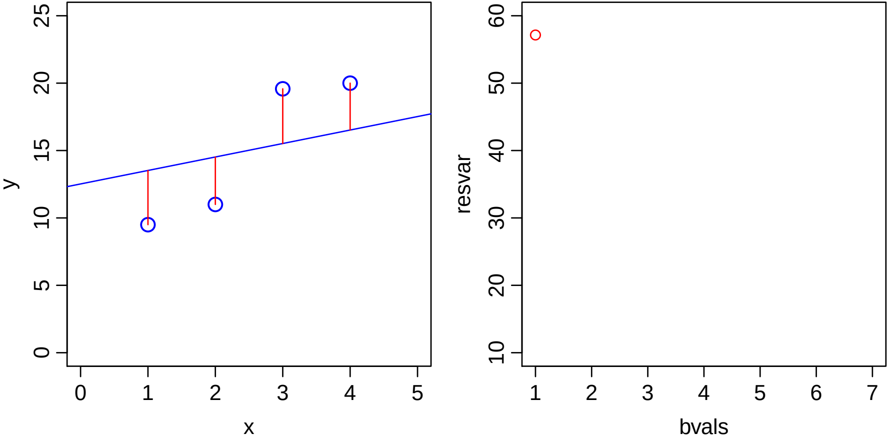
- Consider a predictor x, data y, n observations, and a model that we want to fit to the data:
  $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
  where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ are the model's k parameters

So what — why is intrinsic non-linearity a problem?

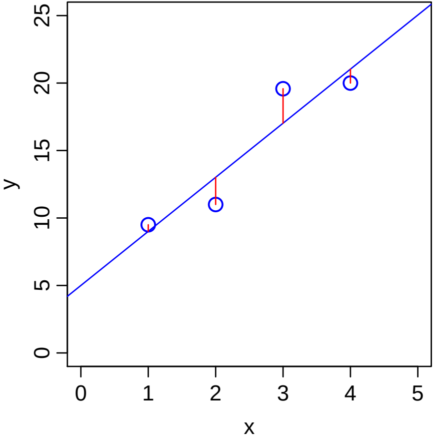
Recall what the Least Squares method does:

- Consider a predictor x, data y, n observations, and a model that we want to fit to the data:
  $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
  where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ are the model's k parameters
- The objective is to find estimates of values of the k parameters ($\hat{\beta}_j$) that minimize the sum (S) of squared residuals (r_i) (AKA RSS):
  $S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$

OLS minimizes the *sum* of the *squared* residuals



If the model is linear, the solution is easy using algebra



$$y_1 = \beta_0 + \beta_1 x_1 + \varepsilon_i$$

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

$$\beta_0 = 5; \beta_1 = 4$$

Intrinsic non-linearity does not allow a algebraic solution

- So, then, in an intrinsically non-linear model such as $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$ the derivatives $\frac{\partial r_i}{\partial \beta_j}$ are naughty

Intrinsic non-linearity does not allow a algebraic solution

- So, then, in an intrinsically non-linear model such as $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$ the derivatives $\frac{\partial r_i}{\partial \beta_j}$ are naughty
- That is, they are functions of both x and the parameters β_j, so the gradient equations do not have a solution like the OLS case

Intrinsic non-linearity does not allow a algebraic solution

- So, then, in an intrinsically non-linear model such as $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$ the derivatives $\frac{\partial r_i}{\partial \beta_j}$ are naughty
- That is, they are functions of both x and the parameters β_j, so the gradient equations do not have a solution like the OLS case
- So the nice trick of solving $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is impossible *mathematically*

So — Enter NLLS

But we can use brute-force computation to find close-to-optimal least squares minimization!

- Choose initial values for the parameters we want to estimate (β_j's)

So — Enter NLLS

But we can use brute-force computation to find close-to-optimal least squares minimization!

- Choose initial values for the parameters we want to estimate (β_j's)
- Then, "refine" the parameters *iteratively* by calculating $\frac{\partial r_i}{\partial \beta_j}$ *approximately* — this approximation is the *Jacobian* (the gradient), which is a matrix of the $\frac{\partial r_i}{\partial \beta_j}$'s

SO — ENTER NLLS

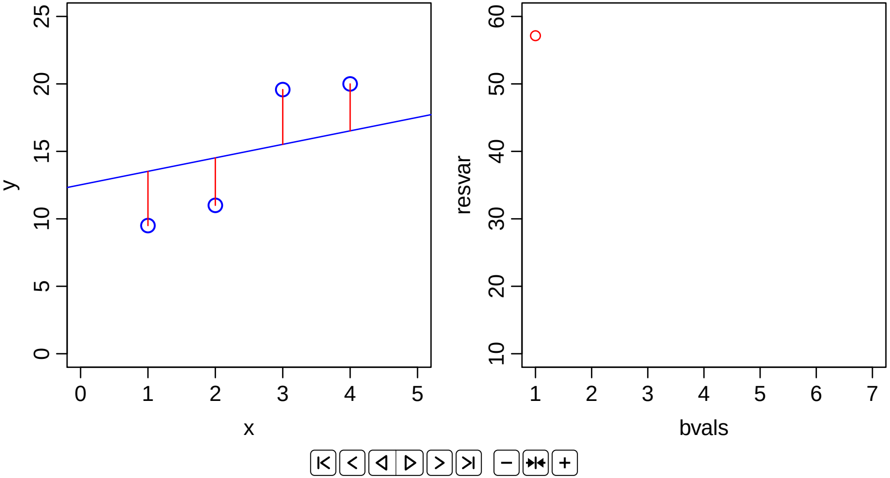
But we can use brute-force computation to find close-to-optimal least squares minimization!

- Choose initial values for the parameters we want to estimate (β_j's)
- Then, "refine" the parameters *iteratively* by calculating $\frac{\partial r_i}{\partial \beta_j}$ *approximately* — this approximation is the *Jacobian* (the gradient), which is a matrix of the $\frac{\partial r_i}{\partial \beta_j}$'s
- Whether a refinement has taken place in any step of the iteration is determined by re-calculating the residuals at that step

So — Enter NLLS

But we can use brute-force computation to find close-to-optimal least squares minimization!

- Choose initial values for the parameters we want to estimate (β_j's)
- Then, "refine" the parameters *iteratively* by calculating $\frac{\partial r_i}{\partial \beta_j}$ *approximately* — this approximation is the *Jacobian* (the gradient), which is a matrix of the $\frac{\partial r_i}{\partial \beta_j}$'s
- Whether a refinement has taken place in any step of the iteration is determined by re-calculating the residuals at that step
- Eventually, if it all goes well, we find a combination of β_j's that is *very close* to the desired solution $\frac{\partial S}{\partial \beta_j} = 0, j = 0, 1, 2, \ldots, k$



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

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

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

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

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

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:

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:
  - Logistic population growth

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:
  - Logistic population growth
  - Allometric growth

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:
  - Logistic population growth
  - Allometric growth
  - Michaelis-Menten biochemical kinetics (two parameters $V_{\max}$ and K_m: $v = \frac{V_{\max}[S]}{K_m+[S]}$

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:
  - Logistic population growth
  - Allometric growth
  - Michaelis-Menten biochemical kinetics (two parameters $V_{\max}$ and K_m: $v = \frac{V_{\max}[S]}{K_m+[S]}$
  - Responses of metabolic rates to changing temperature

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:
    - Logistic population growth
    - Allometric growth
    - Michaelis-Menten biochemical kinetics (two parameters $V_{\max}$ and K_m: $v = \frac{V_{\max}[S]}{K_m+[S]}$
    - Responses of metabolic rates to changing temperature
    - Consumer-Resource (e.g., predator-prey) functional responses

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

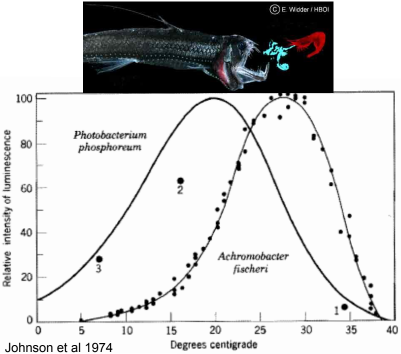
- Many observations in biology are just *not* well-fitted by a linear model
- That is, the underlying biological phenomena/phenomenon are not well-described by a linear equation
- Examples:
    - Logistic population growth
    - Allometric growth
    - Michaelis-Menten biochemical kinetics (two parameters $V_{\max}$ and K_m: $v = \frac{V_{\max}[S]}{K_m+[S]}$
    - Responses of metabolic rates to changing temperature
    - Consumer-Resource (e.g., predator-prey) functional responses
    - Time-series data (e.g., fitting a sinusoidal function)

OK, fine, why would *I* ever need NLLS?

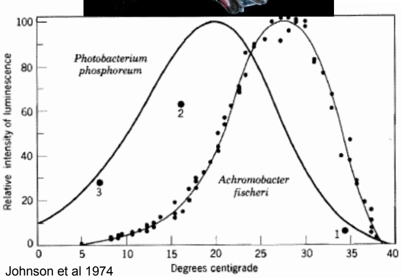
- Many observations in biology are just *not* well-fitted by a linear model
- That is, the underlying biological phenomena/phenomenon are not well-described by a linear equation
- Examples:
  - Logistic population growth
  - Allometric growth
  - Michaelis-Menten biochemical kinetics (two parameters $V_{\max}$ and K_m: $v = \frac{V_{\max}[S]}{K_m+[S]}$
  - Responses of metabolic rates to changing temperature
  - Consumer-Resource (e.g., predator-prey) functional responses
  - Time-series data (e.g., fitting a sinusoidal function)
- *Can you think of some examples?*

Example: Temperature and metabolism

EXAMPLE: TEMPERATURE AND METABOLISM

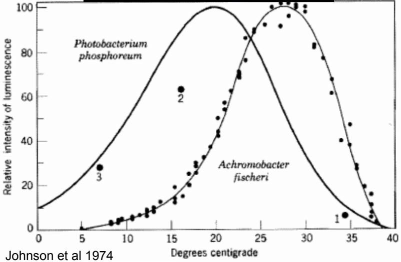


Johnson et al 1974



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

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 vant Hoff 1884, S Arrhenius 1889)

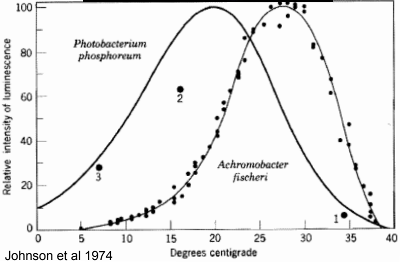


Johnson et al 1974

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

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 vant Hoff 1884, S Arrhenius 1889)

- Surely there is more to thermal responses?
  - Oxygen limitation
  - Complexity of metabolic network
  - Hormonal regulation



Johnson et al 1974

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

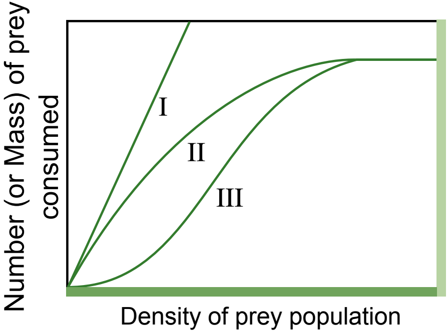
T = temperature (K)
k = Boltzmann constant (eV K^{-1})
E = Activation energy (eV)
T_{pk} = Temperature of peak performance
E_D = Deactivation energy (eV)
(J H vant Hoff 1884, S Arrhenius 1889)

- Surely there is more to thermal responses?
  - Oxygen limitation
  - Complexity of metabolic network
  - Hormonal regulation
- *What about alternative models?*

Example: Functional responses

$$f(x_R) = \frac{a x_R^{q+1}}{1+hax_R^{q+1}}$$ (Holling, 1959)

x_R = Resource density (Mass / Area or Volume)
a = Search rate (Area or Volume / Time)
h = Handling time
q = Shape parameter (dimensionless)



Note that:

- NLLS fitting can yield $h < 0$, $q < 0$, or both
- $h < 0$ is biologically impossible but indicates an upward curving response
- $q < 0$ is biologically unlikely as it indicates a decline in search rate with resource density (but is useful as a measure of deviation away from a type III response)

So the general procedure is:

1. Start with an initial value for each parameter in the model

So the general procedure is:

1. Start with an initial value for each parameter in the model
2. Generate the curve defined by the initial values

NLLS FITTING

So the general procedure 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)

NLLS FITTING

So the general procedure 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

NLLS FITTING

So the general procedure 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)

NLLS fitting

So the general procedure 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

NLLS FITTING

So the general procedure 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

The *tricky part — adjust parameters to make curve come closer to the data points* (step 4) has two main algorithms that you can choose between:

- The Gauss-Newton algorithm is the default in the `nls` package (part of the `stats` base package) — good in many cases, but doesn't work very well if the model is mathematically weird (the optimization landscape is difficult) and the starting values for parameters are far-off-optimal

NLLS FITTING

The *tricky part — adjust parameters to make curve come closer to the data points* (step 4) has two main algorithms that you can choose between:

- The Gauss-Newton algorithm is the default in the `nls` package (part of the `stats` base package) — good in many cases, but doesn't work very well if the model is mathematically weird (the optimization landscape is difficult) and the starting values for parameters are far-off-optimal

- The Levenberg-Marquardt (LM) switches switches between Gauss-Newton and "gradient descent" and is more robust against starting values that are far-off-optimal — available in `R` through the the `minpack.lm` package
  `http://cran.r-project.org/web/packages/minpack.lm`

NLLS FITTING

The *tricky part — adjust parameters to make curve come closer to the data points* (step 4) has two main algorithms that you can choose between:

- The Gauss-Newton algorithm is the default in the `nls` package (part of the `stats` base package) — good in many cases, but doesn't work very well if the model is mathematically weird (the optimization landscape is difficult) and the starting values for parameters are far-off-optimal

- The Levenberg-Marquardt (LM) switches switches between Gauss-Newton and "gradient descent" and is more robust against starting values that are far-off-optimal — available in `R` through the the `minpack.lm` package
  `http://cran.r-project.org/web/packages/minpack.lm`

- The command is `nlsLM`

NLLS fitting

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - R^2

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - R^2
  - Estimated coefficients

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - R^2
  - Estimated coefficients
  - For each coefficient, standard error (can be used for CI's), t-statistic and corresponding (two-sided) p-value

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - R^2
  - Estimated coefficients
  - For each coefficient, standard error (can be used for CI's), t-statistic and corresponding (two-sided) p-value
- The function `summary.nls` will give you all these measures

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
    - Sums of deviations of the data points from the final model fit (final RSS)
    - R^2
    - Estimated coefficients
    - For each coefficient, standard error (can be used for CI's), t-statistic and corresponding (two-sided) p-value
- The function `summary.nls` will give you all these measures
- Remember, the precise parameter values you obtain will depend in part on the initial values chosen and the convergence criteria

NLLS FITTING

- Once the algorithm as converged (hopefully – but you may be surprised how well it usually works), you need to get the goodness of fit measures
- First, of course, examine the fits visually
- Also, report the best-fit results, including:
  - Sums of deviations of the data points from the final model fit (final RSS)
  - R^2
  - Estimated coefficients
  - For each coefficient, standard error (can be used for CI's), t-statistic and corresponding (two-sided) p-value
- The function `summary.nls` will give you all these measures
- Remember, the precise parameter values you obtain will depend in part on the initial values chosen and the convergence criteria
- You may also want to compare multiple models.

NLLS ASSUMPTIONS

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

- No (in practice, minimal) measurement error in explanatory variable (x-axis variable)

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

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 error distribution is Gaussian — for example, what would the error distribution of this non-linear model be: $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$

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 error distribution is Gaussian — for example, what would the error distribution of this non-linear model be: $y_i = \beta_0 e^{\beta_2 x_i} + \varepsilon_i$
- What if the errors are not normal? — use Maximum Likelihood or Bayesian methods instead.

MORE NLLS TIPS

- You can use mixed-effects modelling with NLLS in R; the package is `nlme` `https://stat.ethz.ch/R-manual/R-devel/library/nlme/html/nlme.html` (You are probably stuck with the Gauss-Newton algorithm with `nlme` though)

MORE NLLS TIPS

- You can use mixed-effects modelling with NLLS in R; the package is `nlme` `https://stat.ethz.ch/R-manual/R-devel/library/nlme/html/nlme.html` (You are probably stuck with the Gauss-Newton algorithm with `nlme` though)
- You can also use Python — look up `lmfit` `https://lmfit.github.io/lmfit-py/index.html`. `python` seems to have a better Levenberg-Marqualdt implementation than `R`

Readings and Resources

- Motulsky, Harvey, and Arthur Christopoulos. Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting. OUP USA, 2004.
- Johnson, J. B. & Omland, K. S. 2004 Model selection in ecology and evolution. Trends Ecol. Evol. 19, 101–108.

So what — why is intrinsic non-linearity a problem?

That's all well and good, but we can use maths instead of brute-force computation!

- Our model is $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$

So what — why is intrinsic non-linearity a problem?

That's all well and good, but we can use maths instead of brute-force computation!

- Our model is $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
- We want to find estimates of values of the parameters ($\hat{\beta}_j$) that *minimize* the sum (S) of squared residuals (r_i) (AKA "RSS")
  $S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$

So what — why is intrinsic non-linearity a problem?

That's all well and good, but we can use maths instead of brute-force computation!

- Our model is $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
- We want to find estimates of values of the parameters ($\hat{\beta}_j$) that *minimize* the sum (S) of squared residuals (r_i) (AKA "RSS")
  $S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$
- For this we can solve $\frac{\partial S}{\partial \beta_j} = 0, j = 0, 1, 2, \ldots, k$ to find the *minimum*

So what — why is intrinsic non-linearity a problem?

That's all well and good, but we can use maths instead of brute-force computation!

- Our model is $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
- We want to find estimates of values of the parameters ($\hat{\beta}_j$) that *minimize* the sum (S) of squared residuals (r_i) (AKA "RSS")
  $S = \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 = \sum_{i=1}^{n} r_i^2$
- For this we can solve $\frac{\partial S}{\partial \beta_j} = 0, j = 0, 1, 2, \ldots, k$ to find the *minimum*
- That is, we need to solve $\frac{\partial \sum_{i=1}^{n} r_i^2}{\partial \beta_j} = 0$

SO WHAT — WHY IS INTRINSIC NON-LINEARITY A PROBLEM?

That's all well and good, but we can use maths instead of brute-force computation!

- Our model is $f(x_i, \boldsymbol{\beta}) + \varepsilon_i$
- We want to find estimates of values of the parameters ($\hat{\beta}_j$) that *minimize* the sum (S) of squared residuals (r_i) (AKA "RSS")
  $S = \sum_{i=1}^{n} \left[y_i - f(x_i, \boldsymbol{\beta})\right]^2 = \sum_{i=1}^{n} r_i^2$
- For this we can solve $\frac{\partial S}{\partial \beta_j} = 0, j = 0, 1, 2, \ldots, k$ to find the *minimum*
- That is, we need to solve $\frac{\partial \sum_{i=1}^{n} r_i^2}{\partial \beta_j} = 0$
- Or, $2 \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial \beta_j} = 0$

So what — why is intrinsic non-linearity a problem?

- Thus, solving $2\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial \beta_j} = 0$
  boils down to finding the "gradient" $\frac{\partial r_i}{\partial \beta_j}$

So what — why is intrinsic non-linearity a problem?

- Thus, solving $2\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial \beta_j} = 0$
  boils down to finding the "gradient" $\frac{\partial r_i}{\partial \beta_j}$
- This is not a problem in linear models, because this gradient is fully solvable as the equation is *intrinsically linear*

SO WHAT — WHY IS INTRINSIC NON-LINEARITY A PROBLEM?

- Thus, solving $2\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial \beta_j} = 0$
  boils down to finding the "gradient" $\frac{\partial r_i}{\partial \beta_j}$
- This is not a problem in linear models, because this gradient is fully solvable as the equation is *intrinsically linear*
- That is, the solution of $\frac{\partial r_i}{\partial \beta_j}$ is simple (enough)

So what — why is intrinsic non-linearity a problem?

- For example, if $f(x_i, \boldsymbol{\beta}) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i$

So what — why is intrinsic non-linearity a problem?

- For example, if $f(x_i, \boldsymbol{\beta}) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
- That is, our model is $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ (Linear Regression)

So what — why is intrinsic non-linearity a problem?

- For example, if $f(x_i, \boldsymbol{\beta}) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
- That is, our model is $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ (Linear Regression)
- Then we want to solve
  $\frac{\partial S}{\partial \beta_0} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_0} = 0$
  $\frac{\partial S}{\partial \beta_1} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_1} = 0$

So what — why is intrinsic non-linearity a problem?

- And, solving

  $\frac{\partial S}{\partial \beta_0} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_0} = 0$
  $\frac{\partial S}{\partial \beta_1} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_1} = 0$

  just boils down to solving two simultaneous equations because $\frac{\partial r_i}{\partial \beta_j}$ is simple *because* the model is intrinsically linear:

  $-n\beta_0 + \sum_{i=1}^{n} y_i + \beta_1 \sum_{i=1}^{n} x_i = 0$
  $\sum_{i=1}^{n} x_i y_i - \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = 0$

So what — why is intrinsic non-linearity a problem?

- And, solving
  $\frac{\partial S}{\partial \beta_0} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_0} = 0$
  $\frac{\partial S}{\partial \beta_1} = \sum_{i=1}^{n} \frac{\partial [y_i - (\beta_0 + \beta_1 x_i)]^2}{\partial \beta_1} = 0$
  just boils down to solving two simultaneous equations because $\frac{\partial r_i}{\partial \beta_j}$ is simple *because* the model is intrinsically linear:

  $-n\beta_0 + \sum_{i=1}^{n} y_i + \beta_1 \sum_{i=1}^{n} x_i = 0$
  $\sum_{i=1}^{n} x_i y_i - \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = 0$

- That is, we need to solve $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ (*this is what R solves when you use* `lm()`)