Lecture 4: Optimisation, maximum likelihood and nonlinear least squares in R

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Minimisation, Maximisation, Optimisation . . .

- **Problem:** find minimum or maximum of a function of one or more variables ('optimisation')
- Maximisation of a function h is equivalent to minimisation of $-h$
- **Main statistical applications:** estimation using maximum likelihood & least squares

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

- **Problem:** find minimum or maximum of a function of one or more variables ('optimisation')
- Maximisation of a function h is equivalent to minimisation of $-h$
- **Main statistical applications:** estimation using maximum likelihood & least squares

Assignment Project Exam Help

Standard approach

https://tutorcs.com

1. Differentiate $h(x)$ to get $h'(x)$.
2. Find the solutions (roots) of $h'(x) = 0$.
3. Check that they are in fact minima / maxima.

Practical difficulty: step 2 not always possible

WeChat: cstutorcs

Minimisation, Maximisation, Optimisation . . .

- **Problem:** find minimum or maximum of a function of one or more variables ('optimisation')
- Maximisation of a function h is equivalent to minimisation of $-h$
- **Main statistical applications:** estimation using maximum likelihood & least squares

Assignment Project Exam Help

Standard approach

https://tutorcs.com

1. Differentiate $h(x)$ to get $h'(x)$.
2. Find the solutions (roots) of $h'(x) = 0$.
3. Check that they are in fact minima / maxima.

Practical difficulty: step 2 not always possible

WeChat: cstutorcs

Solution

Find a value for x that is **approximately** a minimum of $h(x)$ using numerical (computational) methods.

Newton-Raphson Method

Assignment Project Exam Help

Idea

- Start with an initial guess x_0.
- Find second-order Taylor approximation to $h(x)$ about $x = x_0$
- Minimise approximation to obtain (hopefully) improved estimate x_1.
  - Approximation is quadratic function $\Rightarrow$ easy to minimise.
- Find second-order Taylor approximation to $h(x)$ about $x = x_1$ ...

https://tutorcs.com

WeChat: cstutorcs

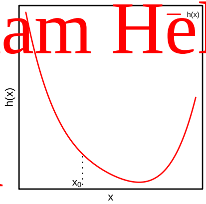
- Algorithm is basis for many optimisation algorithms in modern software packages.

Newton-Raphson for functions of one variable

- Expansion of $h(x)$ around $x = x_0$ is

Assignment Project Exam Help

$$h(x) = h(x_0) + (x - x_0)h'(x_0) + \frac{(x - x_0)^2}{2} h''(x_0) + \dots$$



https://tutorcs.com

WeChat: cstutorcs

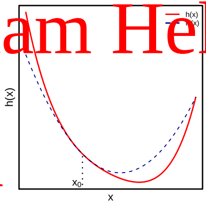
Newton-Raphson for functions of one variable

- Expansion of $h(x)$ around $x = x_0$ is

$$h(x) = h(x_0) + (x - x_0)h'(x_0) + \frac{(x - x_0)^2}{2} h''(x_0) + \dots$$

- So start by working with $H(x) \approx h(x)$, where

$$H(x) = h(x_0) + (x - x_0)h'(x_0) + \frac{(x - x_0)^2}{2} h''(x_0) \; .$$



Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Newton-Raphson for functions of one variable

- Expansion of $h(x)$ around $x = x_0$ is

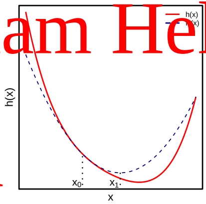
$$h(x) = h(x_0) + (x - x_0)h'(x_0) + \frac{(x - x_0)^2}{2} h''(x_0) + \ldots$$

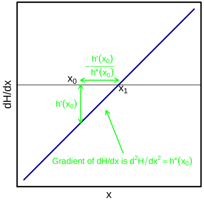
- So start by working with $H(x) \approx h(x)$, where

$$H(x) = h(x_0) + (x - x_0)h'(x_0) + \frac{(x - x_0)^2}{2} h''(x_0) \, .$$

- Solve $dH/dx = 0$ to obtain new value x_1:

$$\frac{dH}{dx} = h'(x_0) + (x - x_0)h''(x_0) \Rightarrow x_1 = x_0 - \frac{h'(x_0)}{h''(x_0)} \, .$$





Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

One step, two steps, many steps . . .

- We can now repeat this process and (hopefully) get an even better approximation x_2 to the minimum / maximum of h
- Result is an iterative process:

$$x_{i+1} = x_i - \frac{h'(x_i)}{h''(x_i)}, \qquad i = 0, 1, 2, \ldots .$$

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

One step, two steps, many steps . . .

- We can now repeat this process and (hopefully) get an even better approximation x_2 to the minimum / maximum of h
- Result is an iterative process:

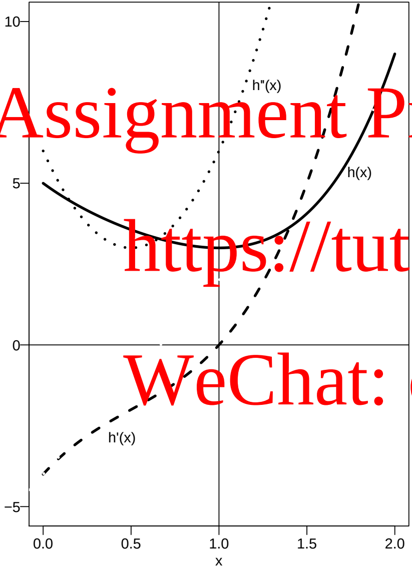
$$x_{i+1} = x_i - \frac{h'(x_i)}{h''(x_i)}, \qquad i = 0, 1, 2, \ldots .$$

Notes on Newton-Raphson

- Need to evaluate $h'(x)$ and $h''(x)$ at any point x
  - In particular, h must be twice continuously differentiable.
- Can also be used for functions of several variables, starting from multivariable Taylor approximation

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

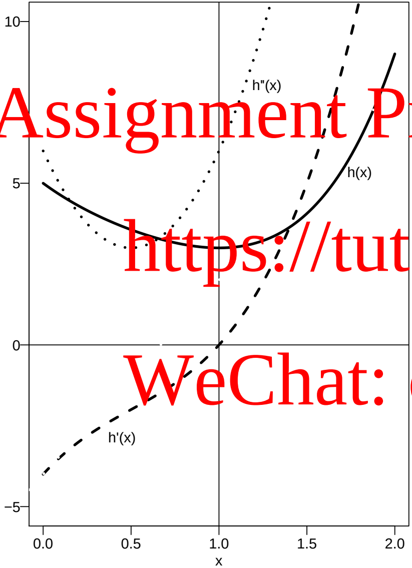
Newton-Raphson example: $h(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$



- Required derivatives are $h'(x) = 4x^3 - 6x^2 + 6x - 4$ and $h''(x) = 12x^2 - 12x + 6$.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Newton-Raphson example: $h(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$

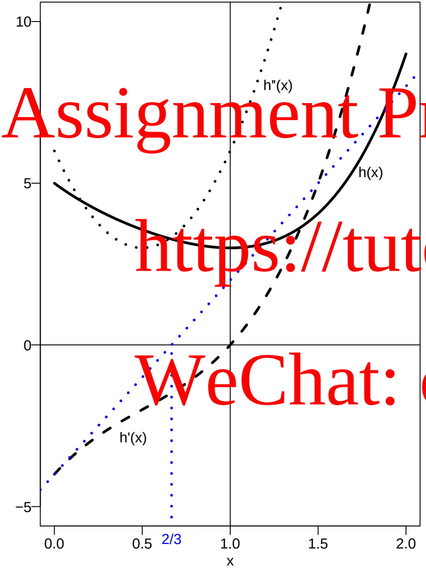


- Required derivatives are $h'(x) = 4x^3 - 6x^2 + 6x - 4$ and $h''(x) = 12x^2 - 12x + 6$.
- Newton-Raphson update formula is

$$x_{i+1} = x_i - \frac{4x_i^3 - 6x_i^2 + 6x_i - 4}{12x_i^2 - 12x_i + 6}.$$

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Newton-Raphson example: $h(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$



Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

- Required derivatives are $h'(x) = 4x^3 - 6x^2 + 6x - 4$ and $h''(x) = 12x^2 - 12x + 6$.
- Newton-Raphson update formula is

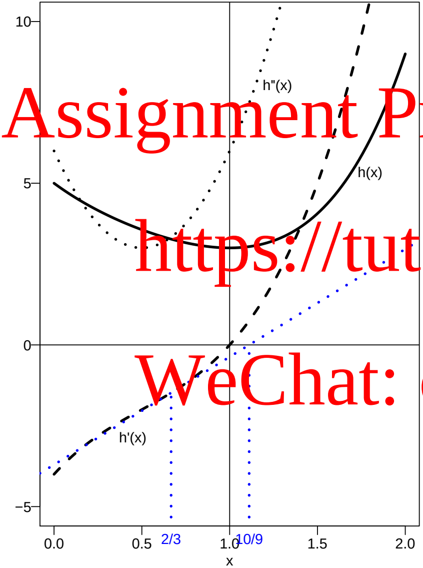
$$x_{i+1} = x_i - \frac{4x_i^3 - 6x_i^2 + 6x_i - 4}{12x_i^2 - 12x_i + 6} \ .$$

- For initial value $x_0 = 0$:

$$x_1 = x_0 - \frac{h'(x_0)}{h''(x_0)} = 0 - \frac{-4}{6} = \frac{2}{3} \ .$$

UCL

Newton-Raphson example: $h(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$



- Required derivatives are $h'(x) = 4x^3 - 6x^2 + 6x - 4$ and $h''(x) = 12x^2 - 12x + 6$.
- Newton-Raphson update formula is
$$x_{i+1} = x_i - \frac{4x_i^3 - 6x_i^2 + 6x_i - 4}{12x_i^2 - 12x_i + 6} \,.$$
- For initial value $x_0 = 0$:
$$x_1 = x_0 - \frac{h'(x_0)}{h''(x_0)} = 0 - \frac{-4}{6} = \frac{2}{3} \,.$$
- Next iteration starts at $x_1 = 2/3$:
$$x_2 = x_1 - \frac{h'(x_1)}{h''(x_1)} = \frac{10}{9} \,.$$
- Etc.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Stopping rules

When to stop? Some common criteria

1. Stop when relative change in h is small: $\left|\frac{h(x_{i+1})-h(x_i)}{h(x_i)}\right| < \varepsilon$;
2. Stop when relative change from x_i to x_{i+1} is small: $\left|\frac{x_{i+1}-x_i}{x_i}\right| < \eta$;
3. Stop when number of iterations i reaches a large value N.

Often stop when any one of these conditions is satisfied: algorithm can be implemented using a `while()` loop;

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

When to stop? Some common criteria

1. Stop when relative change in h is small: $\left|\frac{h(x_{i+1})-h(x_i)}{h(x_i)}\right| < \varepsilon$;
2. Stop when relative change from x_i to x_{i+1} is small: $\left|\frac{x_{i+1}-x_i}{x_i}\right| < \eta$;
3. Stop when number of iterations i reaches a large value N.

Often stop when any one of these conditions is satisfied: algorithm can be implemented using a `while()` loop;

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Further notes on stopping

- ε, η must be specified: e.g. 10^{-6}
- Relative change can be problematic if x_i or $h(x_i)$ is close to zero
  - Modern software usually incorporates measures of absolute change as well: $|h(x_{i+1}) - h(x_i)|$ and $|x_{i+1} - x_i|$.

When to stop? Some common criteria

Assignment Project Exam Help

1. Stop when relative change in h is small: $\left|\frac{h(x_{i+1})-h(x_i)}{h(x_i)}\right| < \varepsilon$;
2. Stop when relative change from x_i to x_{i+1} is small: $\left|\frac{x_{i+1}-x_i}{x_i}\right| < \eta$;
3. Stop when number of iterations i reaches a large value N.

Often stop when any one of these conditions is satisfied: algorithm can be implemented using a `while()` loop;

https://tutorcs.com

Further notes on stopping

WeChat: cstutorcs

- ε, η must be specified: e.g. 10^{-6}
- Relative change can be problematic if x_i or $h(x_i)$ is close to zero
  - Modern software usually incorporates measures of absolute change as well: $|h(x_{i+1}) - h(x_i)|$ and $|x_{i+1} - x_i|$.
- Algorithm may converge to a false minimum (e.g. inflection point), or may not converge at all (e.g. for $h(x) = e^x$) — hence need to check convergence after stopping.

Checking convergence

Assignment Project Exam Help

- Repeat with different initial values x_0 and see if you get the same answer each time.
- If possible plot the function:

https://tutorcs.com

  - Line graph for functions of one variable (`plot()` in R)
  - Colour image or contour map for functions of two variables (`image()` or `contour()` in R)
  - More difficult for functions of 3 or more variables

WeChat: cstutorcs

- Evaluate function over coarse grid of values before using Newton-Raphson, to identify promising regions in which to search for minimum

Minimisation in R: `nlm()`

- **Main commands for numerical optimisation in R**: `nlm()` and `optim()`
  - `nlm()` used in this course.
  - **Basic syntax:** `nlm(`*function name*, *starting value*`)`.
  - R provides default values for other quantities e.g. maximum iteration count, ϵ, η.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Minimisation in R: `nlm()`

- **Main commands for numerical optimisation in R**: `nlm()` and `optim()`
  - `nlm()` used in this course.
  - **Basic syntax:** `nlm(`*function name*, *starting value*`)`.
  - R provides default values for other quantities e.g. maximum iteration count, ϵ, η.
- `nlm` stands for "non-linear minimisation" and uses a "Newton-Raphson-type algorithm".
- `nlm` *estimates* first and second derivatives for each iteration $\Rightarrow$ no need to calculate them yourself ☺

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Minimisation in R: `nlm()`

Assignment Project Exam Help

- **Main commands for numerical optimisation in R**: `nlm()` and `optim()`
  - `nlm()` used in this course.
  - **Basic syntax:** `nlm(`*function name*, *starting value*`)`.
  - R provides default values for other quantities e.g. maximum iteration count, ε, η.

https://tutorcs.com

- `nlm` stands for "non-linear minimisation" and uses a "Newton-Raphson-type algorithm".
- `nlm` *estimates* first and second derivatives for each iteration $\Rightarrow$ no need to calculate them yourself ☺

WeChat: cstutorcs

- If the true Newton-Raphson method converges, it will find either a local maximum or a local minimum (both solve $h'(x) = 0$).
- However, `nlm` always aims to find a (local) *minimum*.

`nlm()` example: minimising $h(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$

Step 1: define an R function to calculate $h(x)$ for any x:

```
> quartic <- function(x) {
+     x^4 - (2*x^3) + (3*x^2) - (4*x) + 5
+ }
```

Assignment Project Exam Help

Step 2: choose an initial guess (say 1.5) and call `nlm()`:

```
> nlm(quartic, 1.5)
$minimum
[1] 3
$estimate
[1] 0.9999992
$gradient
[1] -1.500577e-06
$code
[1] 1
$iterations
[1] 2
```

https://tutorcs.com

WeChat: cstutorcs

Checking convergence with `nlm()`

- `nlm()` returns a `list` result with components minimum (minimised value of $h(x)$), estimate (location of minimum), gradient (value of $h'(x)$ at minimum), code (convergence code — see below) and iterations (# of iterations taken to reach result).

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

Checking convergence with `nlm()`

- `nlm()` returns a `list` result with components minimum (minimised value of $h(x)$), estimate (location of minimum), gradient (value of $h'(x)$ at minimum), code (convergence code — see below) and iterations (# of iterations taken to reach result).

Assignment Project Exam Help

Interpretation of code component

1: Relative gradient is close to zero, result is *probably* reliable (i.e. earlier stopping rule 1).

https://tutorcs.com

2: Successive iterates are within tolerance, result is *probably* reliable (i.e. earlier rule 2).

WeChat: cstutorcs

3: The last step failed to improve on the "estimate", result *may* be reliable.

4: The maximum number of iterations has been reached (same as rule 3), result is unreliable.

5: Large steps have been taken five consecutive times, result is unreliable.

`nlm()`: recommended checks

Assignment Project Exam Help

- Check that `code` takes the value 1 or 2, to confirm that the process has converged.
https://tutorcs.com
- Check that the same result is obtained from different (sensible!) starting points, in case there are local minima.

WeChat: cstutorcs

Assignment Project Exam Help

Reminder: maximum likelihood for independent observations

- Assume $Y_1, \ldots, Y_n$ form an i.i.d. random sample from distribution with pdf / pmf that depends on a parameter θ: $f_Y(y;\theta)$, say, where θ is unknown.

https://tutorcs.com

- Want to use observed sample $y_1, \ldots, y_n$ to estimate θ.
- Maximum likelihood estimate (MLE) is value of θ that maximises the likelihood function defined by $L(\theta;\mathbf{y}) = \prod_{i=1}^{n} f_Y(y_i;\theta)$
  - $\mathbf{y}$ here is vector of all observations: $\mathbf{y} = (y_1\ y_2\ \ldots\ y_n)'$

WeChat: cstutorcs

- Equivalently, MLE maximises the log-likelihood function: $\ell(\theta;\mathbf{y}) = \sum_{i=1}^{n} \log f_Y(y_i;\theta)$.

Example: truncated Poisson distribution

- Let $y_1, \ldots, y_n$ be random sample from truncated Poisson distribution with probability mass function

$$P(Y = y) = \frac{e^{-\theta}\theta^y / y!}{(1 - e^{-\theta})}, \qquad y = 1, 2, \ldots .$$

with $\theta > 0$.

- This is pmf of $Y \mid Y > 0$ where $Y \sim Poi(\theta)$.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Example: truncated Poisson distribution

- Let $y_1, \ldots, y_n$ be random sample from truncated Poisson distribution with probability mass function

$$P(Y = y) = \frac{e^{-\theta}\theta^y / y!}{(1 - e^{-\theta})}, \qquad y = 1, 2, \ldots .$$

  with $\theta > 0$.
  - This is pmf of $Y \mid Y > 0$ where $Y \sim Poi(\theta)$.
- Likelihood function is

$$L(\theta; y) = \prod_{i=1}^{n} \frac{e^{-\theta}\theta^{y_i}}{y_i!\,(1 - e^{-\theta})} = \frac{e^{-n\theta}\,\theta^{\sum_{i=1}^n y_i}}{(1 - e^{-\theta})^n \prod_{i=1}^n y_i!} \qquad \text{(ugh)}.$$

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Example: truncated Poisson distribution

- Let $y_1, \dots, y_n$ be random sample from truncated Poisson distribution with probability mass function

Assignment Project Exam Help

$$P(Y = y) = \frac{e^{-\theta}\theta^y / y!}{(1 - e^{-\theta})}, \qquad y = 1, 2, \dots .$$

with $\theta > 0$.

https://tutorcs.com

  - This is pmf of $Y \mid Y > 0$ where $Y \sim Poi(\theta)$.

- Likelihood function is

WeChat: cstutorcs

$$L(\theta; \mathbf{y}) = \prod_{i=1}^{n} \frac{e^{-\theta}\theta^{y_i}}{y_i!(1 - e^{-\theta})} = \frac{e^{-n\theta}\,\theta^{\sum_{i=1}^n y_i}}{(1 - e^{-\theta})^n \prod_{i=1}^n y_i!} \qquad \text{(ugh)}.$$

- Log-likelihood: $\ell(\theta; \mathbf{y}) = -n\theta + \sum_{i=1}^n y_i \log\theta - n\log(1 - e^{-\theta}) + K$

$$= -n\left\{\theta - \bar{y}\log\theta + \log(1 - e^{-\theta})\right\} + K,$$

where K denotes terms that depend on $\mathbf{y}$ but not θ.

Specimen data:

Value	1	2	3	4	5	6	> 6
Frequency	1486	694	195	37	10	1	0

Defining the data in R (**NB** use of `rep()` to replicate each value the required number of times):

```
> TPdata <- rep(1:6,c(1486,694,195,37,10,1))
> mean(TPdata)
[1] 1.511762
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Truncated Poisson example: MLE in R

Specimen data:

Value	1	2	3	4	5	6	>6
Frequency	1486	694	195	37	10	1	0

Assignment Project Exam Help

Defining the data in R (**NB** use of `rep()` to replicate each value the required number of times):

https://tutorcs.com

```
> TPdata <- rep(1:6,c(1486,694,195,37,10,1))
> mean(TPdata)
[1] 1.511762
```

WeChat: cstutorcs

- To fit truncated Poisson distribution, can maximise log-likelihood ignoring constant terms:

$$-n\left\{\theta-\bar{y}\log\theta+\log(1-e^{-\theta})\right\}$$

- . . . or minimise negative log-likelihood:

$$n\left\{\theta-\bar{y}\log\theta+\log(1-e^{-\theta})\right\}$$

Step 1: R function to calculate $-\ell(\theta; \mathbf{y})$

```
tploglik <- function(theta,y) {
#
# Returns the negative log likelihood of a truncated
# Poisson distribution, up to a constant. Arguments:
#
# theta     Parameter of the distribution
# y         Vector of observations
#
  n <- length(y)
  ybar <- mean(y)
  n*(theta - (ybar*log(theta)) + log(1-exp(-theta)) )
}
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

- **Note:** $-\ell(\theta; \mathbf{y})$ is a function of θ but also depends on data $\mathbf{y}$.
- **So:** `tploglik()` is a function of `theta` but data `y` must be supplied as well.

Step 2: choose initial value and call `nlm()`

```
> nlm(tploglik,1,y=TPdata)
$minimum
[1] 1302.179

$estimate
[1] 0.8924956

$gradient
[1] 2.046863e-06

$code
[1] 1

$iterations
[1] 5
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

- **Code is 1** so result is probably reliable: function converges to MLE $\hat{\theta}$ =0.8925 within 5 iterations.
- **NB** also: `nlm()` gives warnings (not shown) due to trying negative values of `theta` ($\ell(\theta;\mathbf{y})$ involves $\log\theta$)

Another application: nonlinear least squares estimation

Assignment Project Exam Help

- **Recall:** in a linear regression model $Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_j$, the least squares coefficient estimates minimise the sum of squared errors

$$\sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

https://tutorcs.com

  - Explicit formula for least squares estimates is available because the regression function is linear in the coefficients $\{\beta_j\}$.

WeChat: cstutorcs

Another application: nonlinear least squares estimation

- **Recall:** in a linear regression model $Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_j$, the least squares coefficient estimates minimise the sum of squared errors

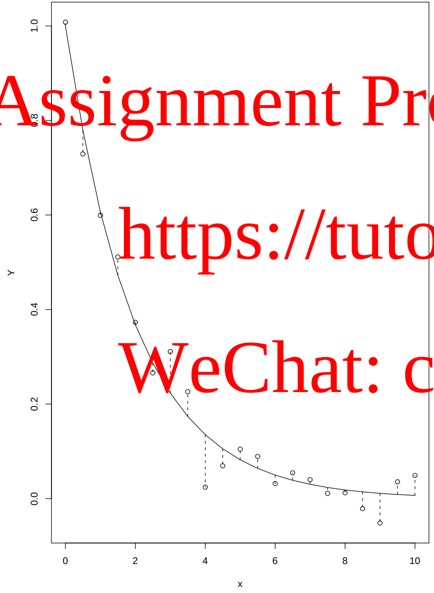
$$\sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

  - Explicit formula for least squares estimates is available because the regression function is linear in the coefficients $\{\beta_j\}$.

- What about nonlinear models? E.g. $Y_i = \beta_0 \, e^{\beta_1 x_i} + \varepsilon_i$ ($i = 1, \ldots, n$) with the $\{\varepsilon_i\}$ iid $N(0, \sigma^2)$.
  - Can't take logs to get a linear model of Y on x because error term would not be additive.
  - Least-squares estimates of β_0 & β_1 minimise $\sum_{i=1}^{n} \left(Y_i - \beta_0 \, e^{\beta_1 x_i} \right)^2$.
  - No explicit formula available $\Rightarrow$ use numerical minimisation.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Non-linear Least Squares: an example



Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

- **Model:**
  $Y_i = \beta_0 \, e^{\beta_1 x_i} + \varepsilon_i (i = 1, \ldots, n)$.
- **Error sum of squares:**
  $\sum (Y_i - \beta_0 e^{\beta_1 x_i})^2$.
- Want to minimise this wrt β_0 and β_1
- Define parameter vector $\theta = (\beta_0 \; \beta_1)'$: then want to minimise sum of squares wrt θ

Nonlinear least squares example: implementation in R

- Suppose data are stored in data frame called `nonlindata`, with columns `x` and `Y`

Assignment Project Exam Help

Step 1: R function to calculate sum of squared errors

```
> sumsqerr <- function(theta,x,Y) {
+   beta0 <- theta[1]
+   beta1 <- theta[2]
+   mu <- beta0*exp(beta1*x)
+   sum((Y-mu)^2)
+ }
```

https://tutorcs.com
WeChat: cstutorcs

Nonlinear least squares example: implementation in R

- Suppose data are stored in data frame called `nonlindata`, with columns `x` and `Y`

Assignment Project Exam Help

Step 1: R function to calculate sum of squared errors

```
> sumsqerr <- function(theta,x,Y) {
+   beta0 <- theta[1]
+   beta1 <- theta[2]
+   mu <- beta0*exp(beta1*x)
+   sum((Y-mu)^2)
+ }
```

https://tutorcs.com

WeChat: cstutorcs

Step 2: choose initial value e.g. $\beta_0 = 1$, $\beta_1 = 2$, and call `nlm()`

```
> nlm(sumsqerr,c(1,2),x=nlindata$x,Y=nlindata$Y)
```

Administrative matters

ICA1 starts next week!!!

- **Two components**: Moodle quiz (25%) and take-home assignment (75%)
- **Moodle quiz:**
  - Questions taken from quizzes for weeks 1–4
  - Quiz takes place in first hour of your timetabled workshop session: you **MUST** turn up to the correct session or you won't be able to take the quiz.
  - You **MUST** use a UCL computer to do the quiz.
  - **Location**: most students in usual workshop location but a small number in Room 316, 26 Bedford Way or in Room 2.23, Chadwick Building **CHECK YOUR EMAIL!!!**
- **Take-home assignment:**
  - Due on Monday 18th February
  - No groupwork (but different questions for everyone)

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs