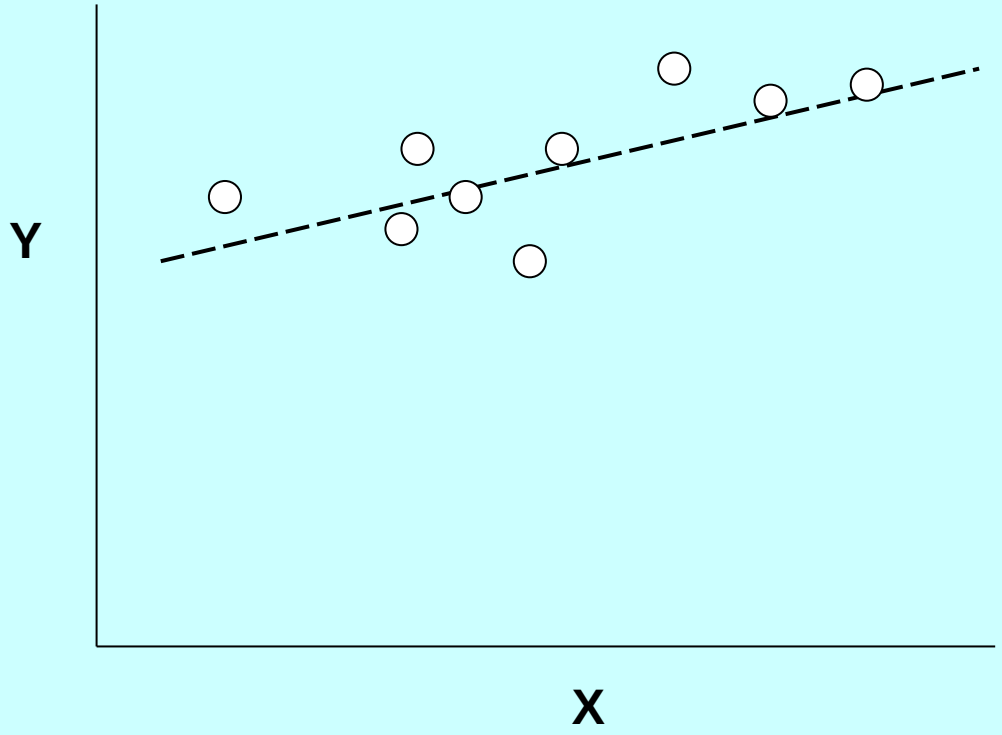
Regression, least squares

Joe Felsenstein

Department of Genome Sciences and Department of Biology

Fitting a straight line

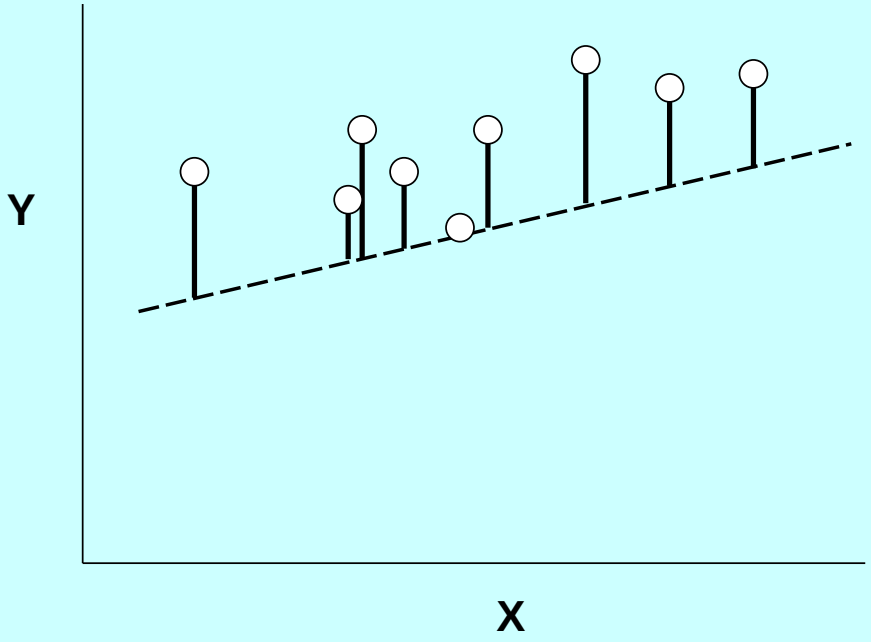


Two distinct cases:

- The X values are chosen arbitrarily by you, and then Y values are measured for each.
- (X, Y) pairs have a joint distribution and are sampled by you.

These are different. **We will assume the former**.

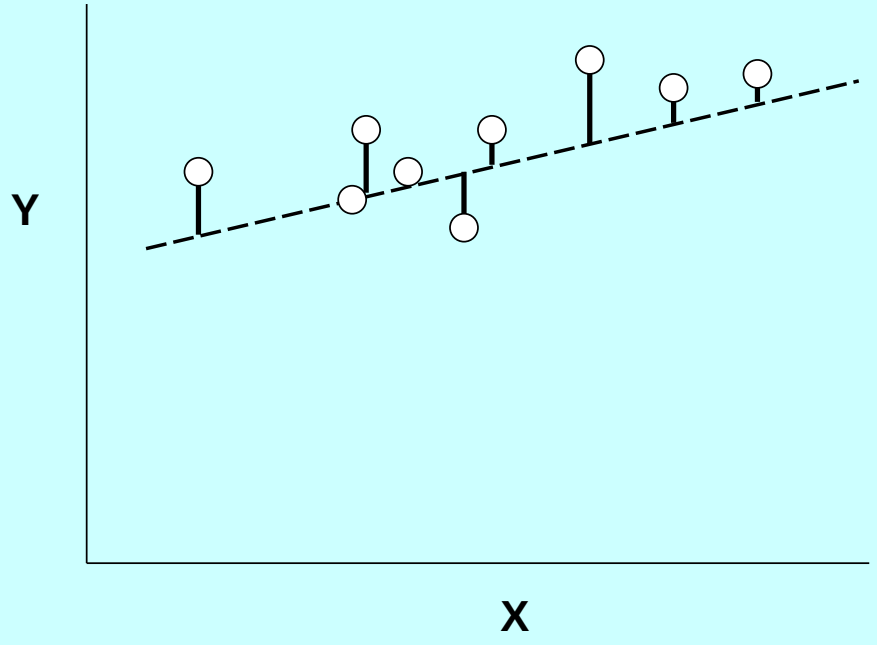
Least squares



Find intercept (a) and slope (b) by minimizing the sum of squares of departures of points from the line:

$$Q = \sum_{i=1}^{n} (Y_i - (a + b X_i))^2$$

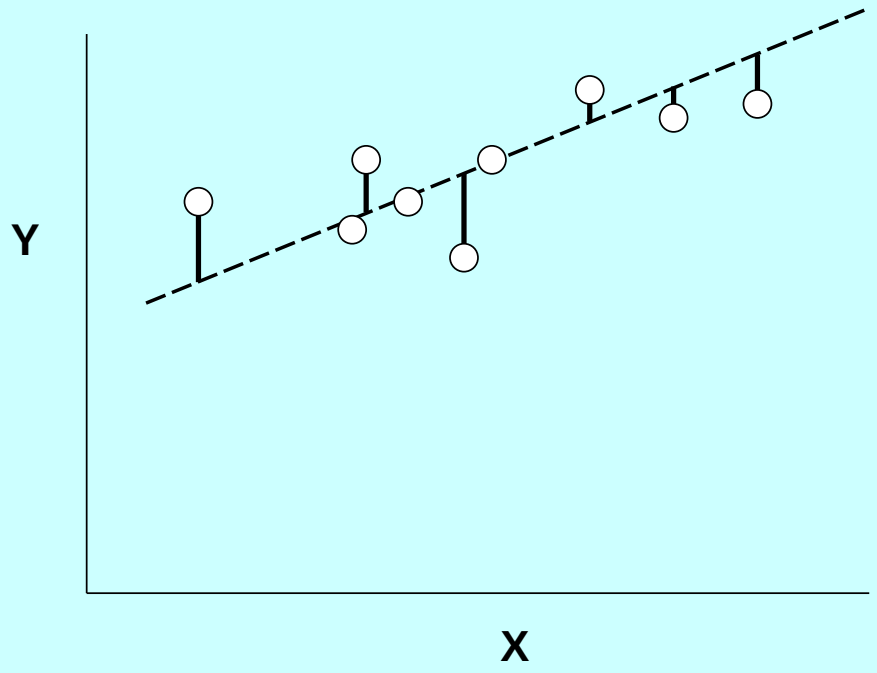
Least squares



Find intercept (a) and slope (b) by minimizing the sum of squares of departures of points from the line:

$$Q = \sum_{i=1}^{n} (Y_i - (a + b X_i))^2$$

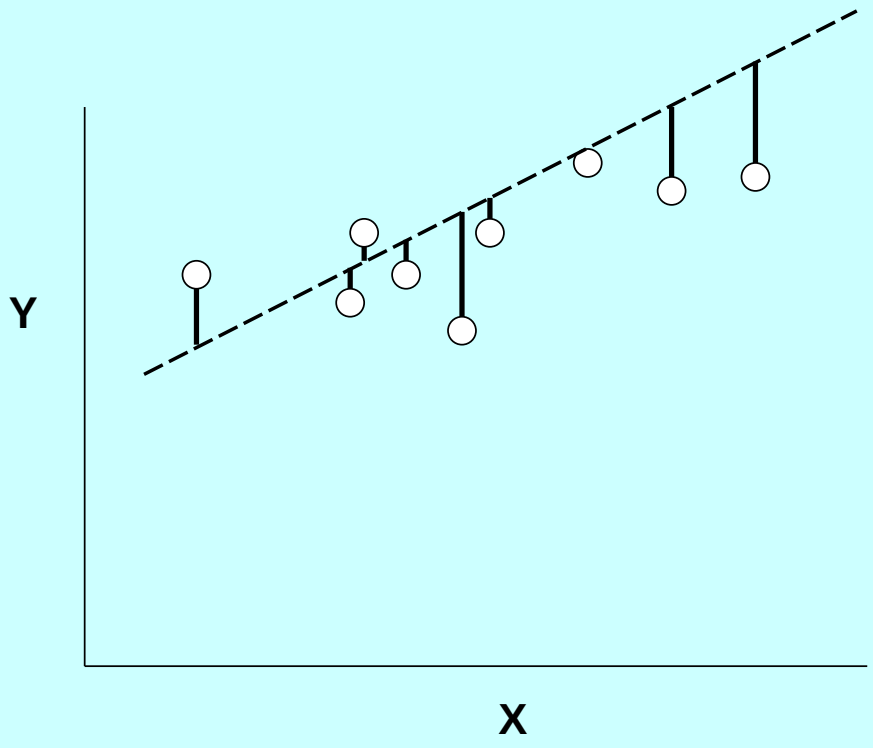
Least squares



Find intercept (a) and slope (b) by minimizing the sum of squares of departures of points from the line:

$$Q = \sum_{i=1}^{n} (Y_i - (a + b X_i))^2$$

Least squares



Find intercept (a) and slope (b) by minimizing the sum of squares of departures of points from the line:

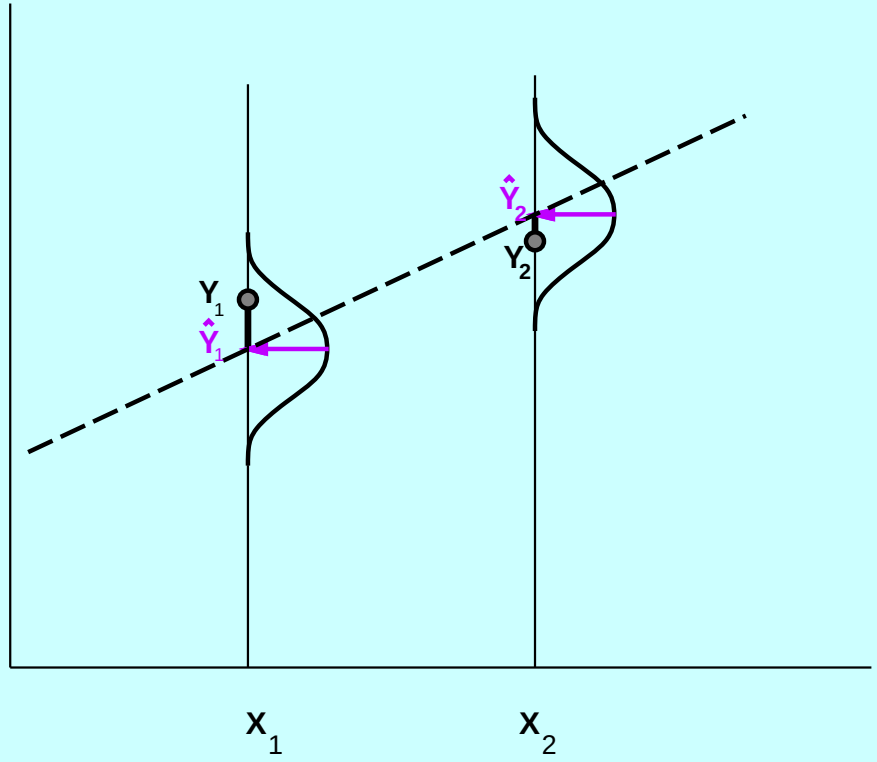
$$Q = \sum_{i=1}^{n} (Y_i - (a + b X_i))^2$$

The least squares solution

- The line passes through the point which is the means of both variables: $(\bar{X}, \bar{Y})$
- Its slope is

$$b_{Y.X} = \frac{\sum_i \left(Y_i - \bar{Y}\right)\left(X_i - \bar{X}\right)}{\sum_i \left(X_i - \bar{X}\right)^2}$$

This assumes



- The X values are known precisely, **without error**.
- Each Y is drawn from a normal distribution whose mean is on the true line
- The standard deviations of these normal distributions are all equal.
- The residuals of the points are independent.

How could we check these in R*?*

What if variances around line are unequal?

When standard deviations around the line are not equal, and are a known function of X (up to a constant anyway), such as $\sigma_i = X^{1/2}$, we can cope with that by having unequal weights of the points.

If the quantities we square are the residuals, expressed as proportions of the (local) standard deviation:

$$Q = \sum_i \left(\frac{Y_i - (a + b\, X_i)}{\sigma_i} \right)^2 = \sum_i \left(\frac{1}{\sigma_i^2} \right) (Y_i - (a + b\, X_i))^2$$

so that the natural weight is the reciprocal of the local variance. This leads to formulas for the slope that weight each term.

Estimating the standard deviation around the line

The variance σ^2 is estimated simply by s^2, the mean square of the deviation from the estimated (unweighted) regression line. But the number of degrees of freedom in the denominator should be $n-2$ as both a and b are being estimated from these data.

$$s^2 = \frac{\sum_i (Y_i - (a + b\,X_i))^2}{n-2}$$

How to do it in R

You can fit a least-squares regression using the function

```
mm <- lsfit(X,Y)
```

where X and Y are vectors of the same length, so the points we fit are

```
(X[1],Y[1]), (X[2],Y[2]), ...
```

The coefficients of the fit are then given by

```
mm$coef
```

The residuals are

```
mm$residuals
```

and to print out the tests for zero slope just do

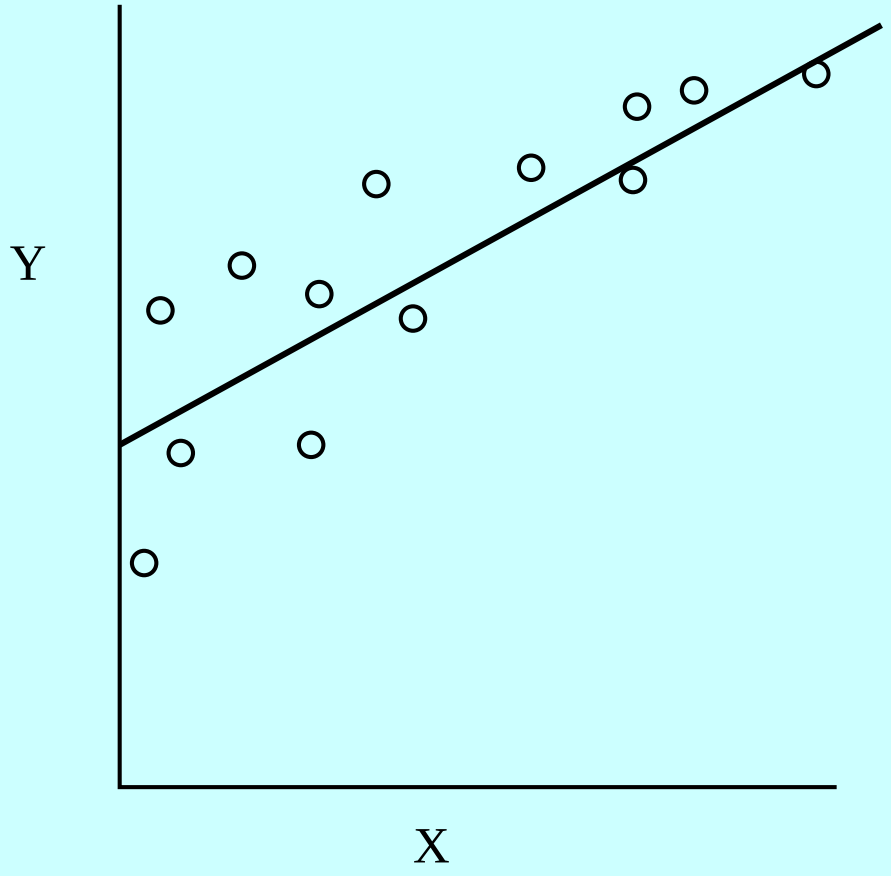
```
ls.print(mm)
```

and you will get something like this:

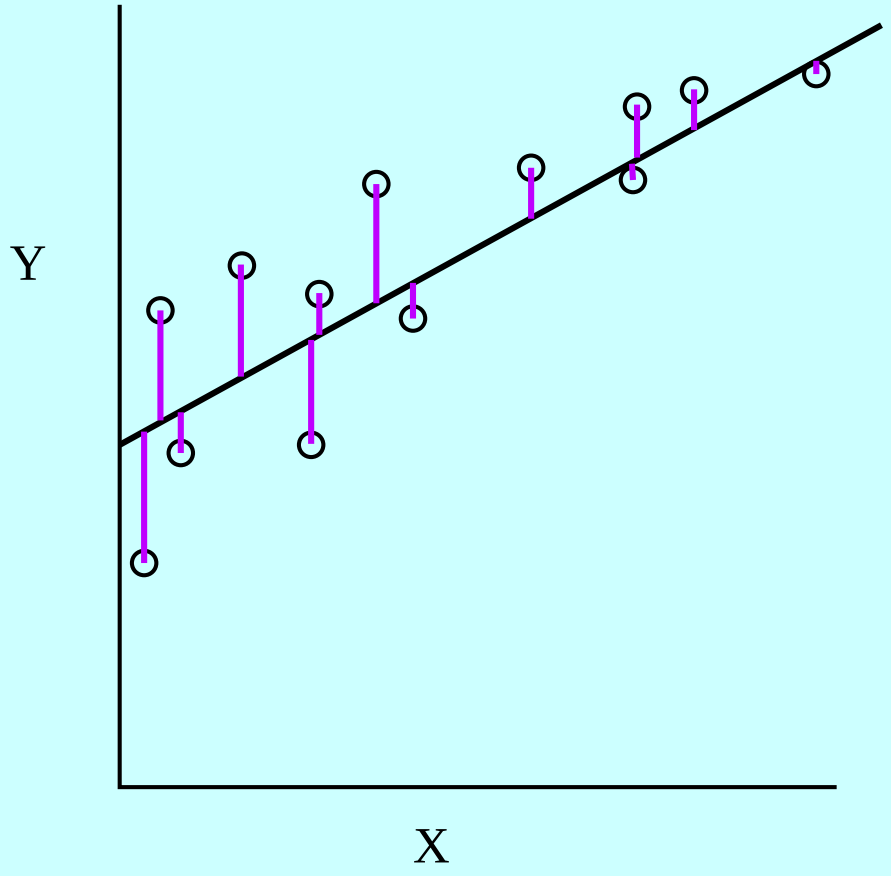
```
Residual Standard Error=12.1645
R-Square=0.1451
F-statistic (df=1, 15)=2.5468
p-value=0.1314

             Estimate  Std.Err  t-value  Pr(>|t|)
 Intercept   51.6283   7.9637   6.4829    0.0000
 X           -0.3031   0.1900  -1.5959    0.1314
```

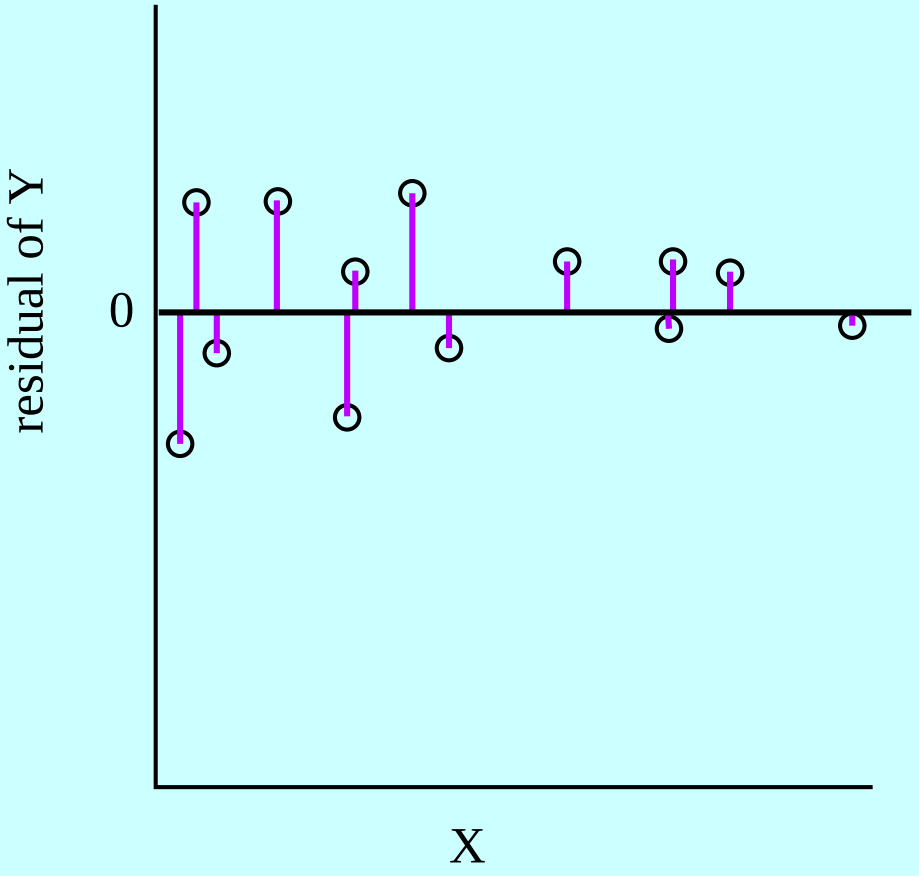
Plotting residuals can be important



Plotting residuals can be important

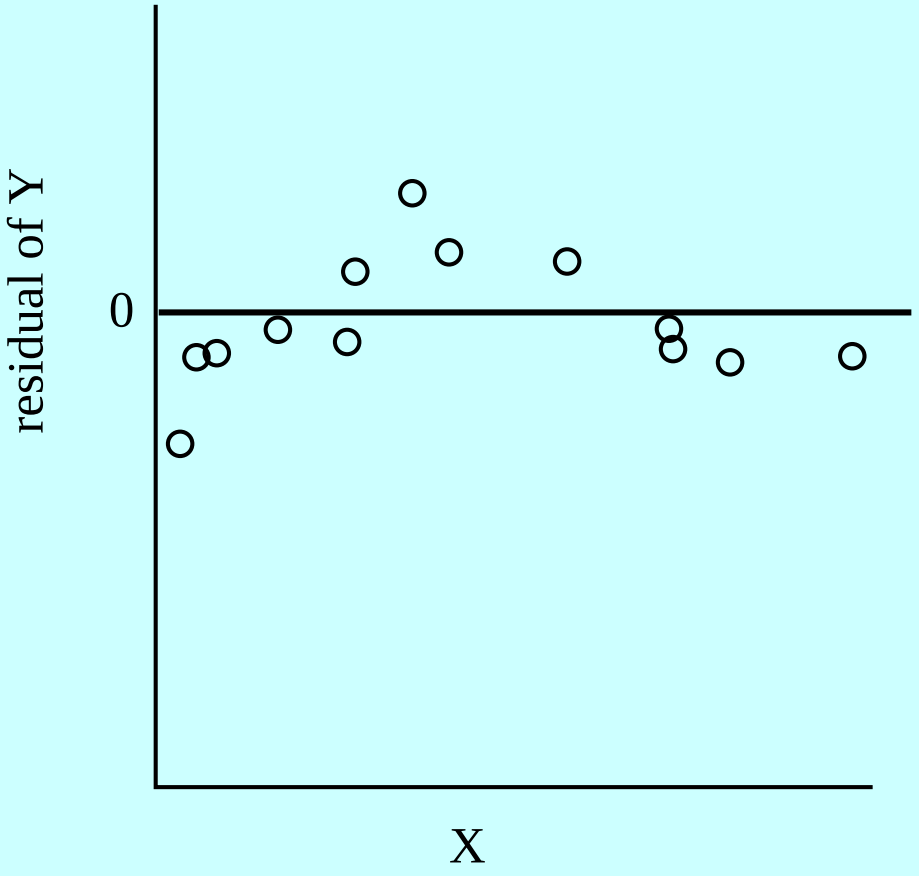


Plotting residuals can be important



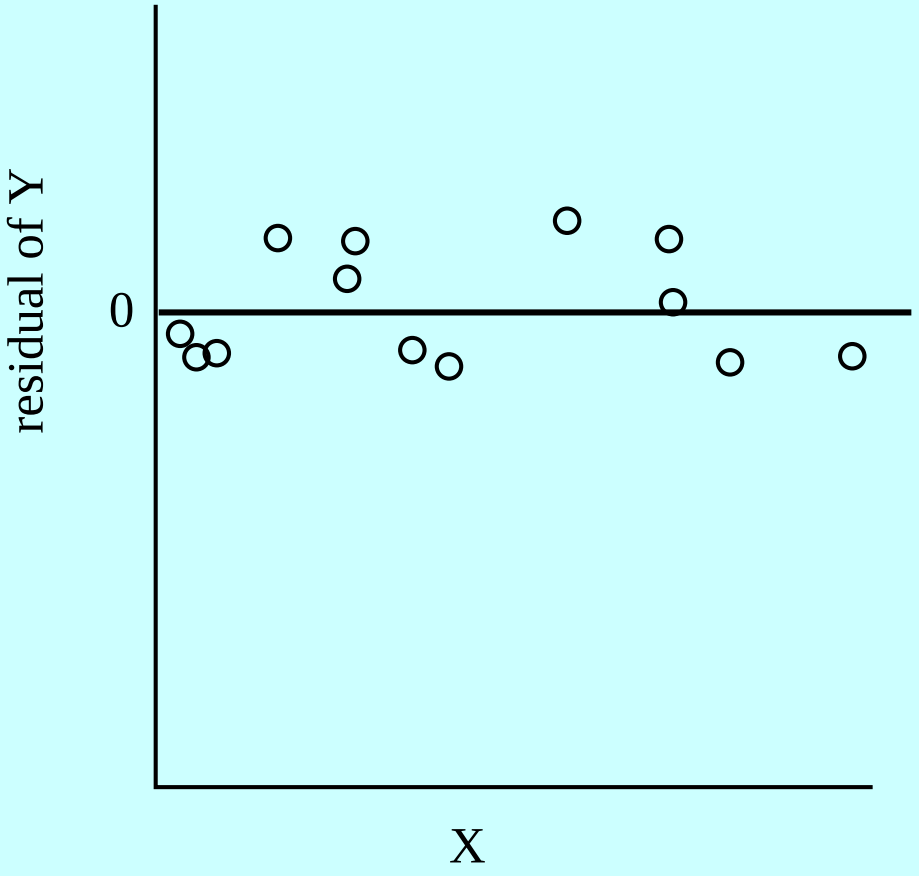
In this case, there is some sign that variances around the line are not the same for all values of X

Plotting residuals can be important, part 2



In this case, the relationship may not be a straight line.

Plotting residuals can be important, part 3



In this case, there is some indication that the residuals are correlated.

Putting confidence limits on the slope

We often want confidence limits on the slope, and to do tests of hypotheses such as that the slope is zero.

The departure of the estimate $b_{y.X}$ from its true value β can be shown to be distributed as

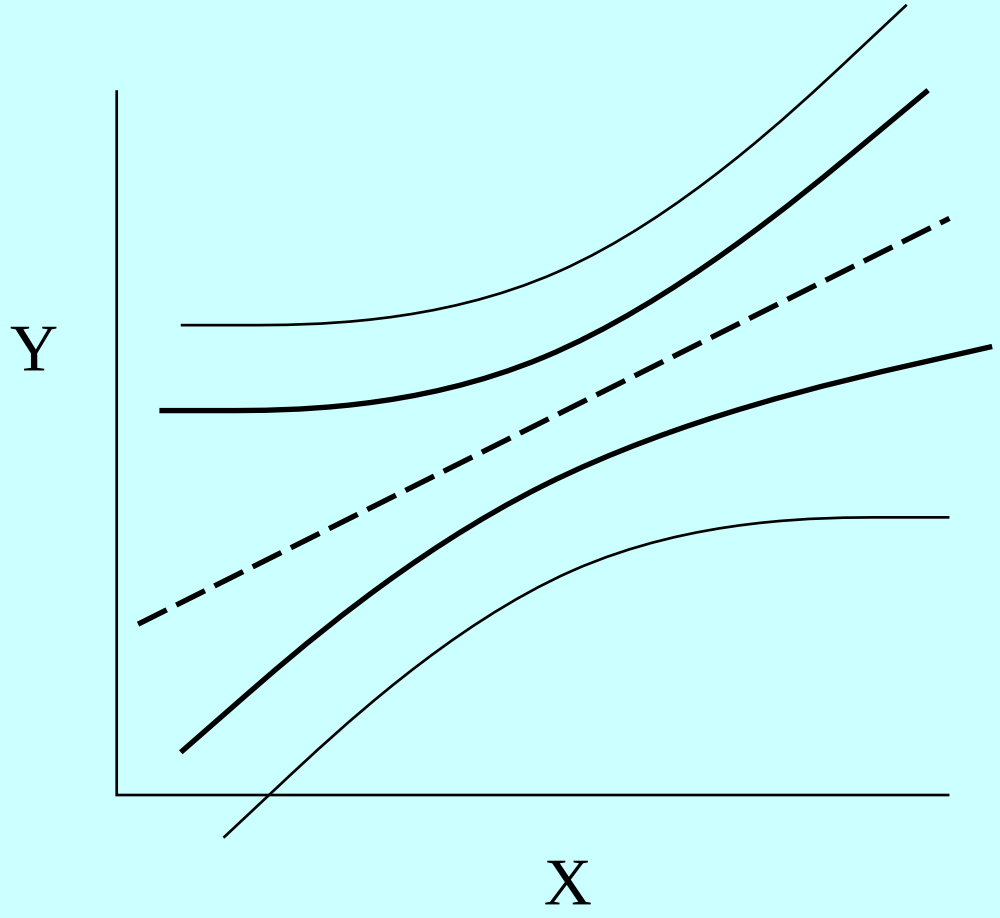
$$b_{Y.X} - \beta \;=\; \frac{t_{n-2}\; s}{\sqrt{\sum_i \left(X_i - \bar{X}\right)^2}}$$

where t_{n-2} is drawn from a t-distribution with $n-2$ degrees of freedom. By plugging in the upper and lower (say) 2.5% points of the t-distribution we can get the confidence limits from this, and test hypotheses such as that $\beta = 0$.

Note that the denominator of the right-hand side implies the sensible point that choosing X's that are far apart helps.

Confidence interval on the line, new points

You can also use the same machinery (in ways not shown here) to put confidence limits on where the line is at any given X (dark curve below) and confidence limits on what a new data point Y will be at that point (wider lighter curves). These are hyperbolas. Note the increasing uncertainty as you extrapolate.



Regression through the origin

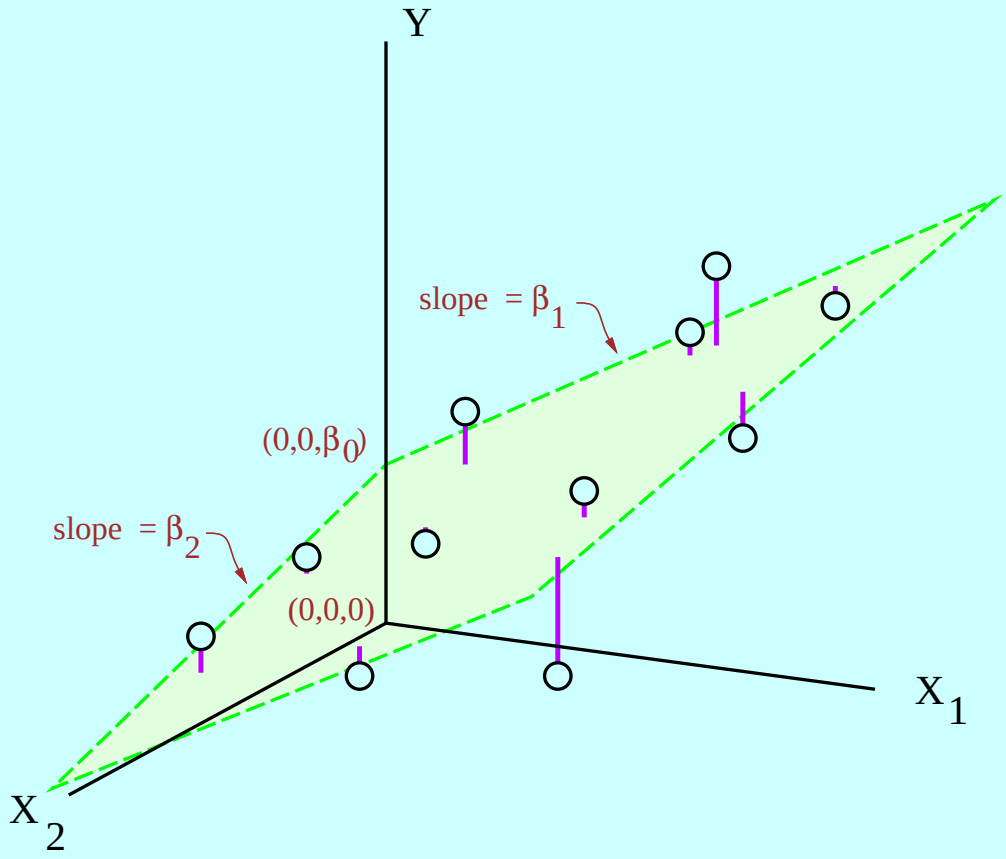
If the regression line must pass through $(0, 0)$, this just means that we replace $\bar{X}$ and $\bar{Y}$ by zero.
The result is that the slope is simply

$$b_{Y.X} = \frac{\sum_i Y_i \, X_i}{\sum_i X_i^2}$$

Multiple regression

One can do linear regression with more than one X variable:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$



Least squares fit for multiple regression

The fit for multiple regression is a least squares fit, to minimize the sum of squared errors of prediction of Y

$$Q = \sum_i (Y_i - (\beta_0 + \beta_1 X_1 + \beta_2 X_2))^2$$

There are weighted versions of this too, when the error variance is different for different values of X_1 and X_2

When we solve for the β_i by differentiating Q and equating the derivatives to zero, we get ..

The equations for multiple regression

You can do regression with one Y and multiple different X variables. This most easily done with matrices (sorry).

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} & X_{13} \\ 1 & X_{21} & X_{22} & X_{23} \\ & \vdots & \vdots & \\ 1 & X_{n1} & X_{n2} & X_{n3} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \text{errors}$$

or, in matrix notation

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

which can be shown to lead to the estimate of the regression coefficients

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^t\mathbf{X}\right)^{-1}\left(\mathbf{X}^t\mathbf{Y}\right)$$

Fitting a (multivariate) linear model

In R there is a function `lm` that uses a formula, leaving out the coefficients and with an implied intercept. So to fit

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

You simply have vectors `X1` and `X2`, and leaving out the term for the intercept and the coefficients, just do:

```
> lm(Y ~ X1+X2)

Call:
lm(formula = Y ~ X1 + X2)

Coefficients:
(Intercept)           X1           X2
   35.60689      0.77367     -0.01474
```

Fitting a polynomial curve

The machinery for fitting multiple regression can be used to fit polynomials – all we have to do is consider $X, X^2, X^3, i \ldots$ to be separate variables. The least-squares equation is, for example:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$

This can be done simply with the R linear model function `lm` this way:

```
> lm(Y ~ poly(X1,degree=3,raw=TRUE))
Call:  lm(formula = Y ~ poly(X1, degree = 3, raw = TRUE))
Coefficients:
                     (Intercept)  poly(X1, degree = 3, raw = TRUE)1
                       16.4896082                          2.9336642
poly(X1, degree = 3, raw = TRUE)2  poly(X1, degree = 3, raw = TRUE)3
                       -0.0823165                          0.0006248
```