



# Canadian Bioinformatics Workshops

[www.bioinformatics.ca](http://www.bioinformatics.ca)

Creative Commons

This page is available in the following languages:  
 Afrikaans Ӏurraposi Catala Dansk Deutsch EӀinyw6 English English (CA) English (GB) English (US) Esperanto  
 Castellano Castellano (AR) Espaol (CL) Castellano (CO) Espaol (Ecuador) Castellano (MX) Castellano (PE)  
 Eestiira Suomieta Franais Franais (CA) Galego In-ua Innelele Itaglian Italiano 日本語 ភាសាខ្មែរ Maori Maori Melayu  
 Nederlands Norsk Sasotho sa Leboa polski Portugala romnani slovenski jezik cрnosrpski (latinnca) Sotho sveneka  
 中文 粵語 (台灣) isiZulu



Attribution-Share Alike 2.5 Canada

**You are free:**

-  **to Share** — to copy, distribute and transmit the work
-  **to Remix** — to adapt the work



**Under the following conditions:**

-  **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
-  **Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar licence to this one.
  - For any reuse or distribution, you must make clear to others the licence terms of this work.
  - Any of the above conditions can be waived if you get permission from the copyright holder.
  - The author's moral rights are retained in this licence.

Disclaimer

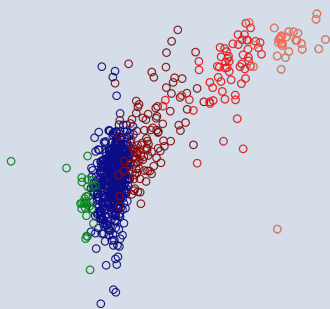
Your fair dealing and other rights are in no way affected by the above.  
 This is a human-readable summary of the Legal Code (the full licence) available in the following languages:  
 English French

[Learn how to distribute your work using this licence](#)

## Module 7 Regression

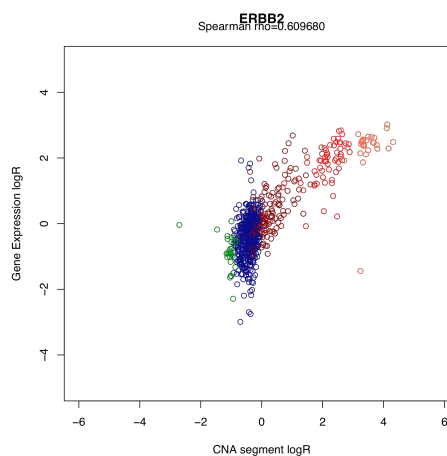


Sohrab Shah  
Exploratory Data Analysis and Essential Statistics using R  
May 6-7, 2010



## Regression

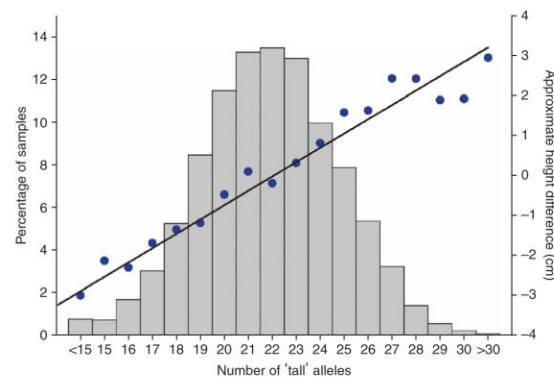
- What is regression?
- Describes how one variable (or set of variables) depend on another variable (or set of variables)
- Example:
  - height vs weight
  - Gene dosage vs expression
  - Survival analysis (Clinical Genomics Workshop)



## Regression

- How is regression different than classification?
  - Regression aims to predict a response that could take on infinite values. Recall that classification is discrete
  - Often characterized as a quantitative prediction rather than qualitative
    - What will the value of my stock be tomorrow?
    - What is the prognosis of a patient given a gene expression signature?
    - If I sequence a genome to 30x coverage, how many SNPs will I detect?

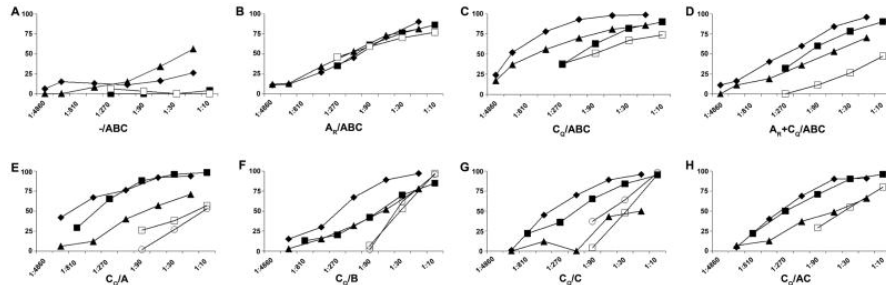
## Examples



The combined impact of the 20 SNPs with a  $P < 5 \times 10^{-7}$ . Subjects were classified according to the number of "tall" alleles at each of the 20 SNPs; the mean height for each group is plotted (blue dots). The black line is a linear regression line through these points. The gray bars represent the proportion of the sample with increasing numbers of "tall" alleles. The approximate height difference (cm) was obtained by multiplying the mean Z-score height for each group by 6.82 cm (the approximate average s.d. of height across the samples used in this study).

From: Weedon MN, et al. Genome-wide association analysis identifies 20 loci that influence adult height. Nat Genet. 2008 May;40(5):575-83. Nat Genet. 2008 May; 40(5): 575-583.

## Examples



Dose-response neutralization curves of individual immune rabbit sera. The serum from one rabbit from each group listed in Table 1 is shown in each panel. Neutralization curves are shown for viruses DJ263 (subtype AG) ( $\nabla$ ), BZ167 (subtype B) ( $\blacksquare$ ), BX08 (subtype B) ( $\blacktriangle$ ), VI313 (subtype A) ( $\square$ ), and 93MW960 (subtype C) ( $\circ$ ). Serum dilutions are shown on the x-axis and percent neutralization is shown on the y-axis. Percent neutralization is based on the activity in the immune sera vs. the corresponding animal's pre-immune serum at the same dilution. Data shown are from one of two or more experiments. The sera used were from the following rabbits: Panel A = rabbit 18; B = rabbit 21; C = rabbit 24; D = rabbit 26; E = rabbit 43; F = rabbit 50; G = rabbit 55; H = rabbit 58.

From: Virology. 2009 Sep 15;392(1):82-93. **Cross-clade neutralizing antibodies against HIV-1 induced in rabbits by focusing the immune response on a neutralizing epitope.** Zolla-Pazner S et al.

Module 7: Regression

bioinformatics.ca

## Types of regression

Linear regression assumes a particular model:

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

$x_i$  is the *independent variable*. Depending on the context, also known as a "predictor variable," "regressor," "controlled variable," "manipulated variable," "explanatory variable," "exposure variable," and/or "input variable."

$y_i$  is the *dependent variable*, also known as "response variable," "regressand," "measured variable," "observed variable," "responding variable," "explained variable," "outcome variable," "experimental variable," and/or "output variable."

$\varepsilon_i$  are "errors" - not in the sense of being "wrong", but in the sense of creating deviations from the idealized model. The  $\varepsilon_i$  are assumed to be independent and  $N(0, \sigma^2)$  (normally distributed), they can also be called *residuals*.

This model has two parameters: the *regression coefficient*  $\beta$ , and the *intercept*  $\alpha$ .

Module 7: Regression

bioinformatics.ca

## Linear regression

- Assumptions:
  - Only two variables are of interest
  - One variable is a response and one a predictor
  - No adjustment is needed for confounding or other between-subject variation
  - Linearity
  - $\sigma^2$  is constant, independent of  $x$
  - $\varepsilon_i$  are independent of each other
  - For proper statistical inference (CI, p-values),  $\varepsilon_i$  are normal distributed

## Linear regression

Linear regression analysis includes:

- estimation of the parameters;
- characterization how good the model is.

## Linear regression: estimation

Parameter estimation: choose parameters that come as close as possible to the "true" values.

Problem: how do we distinguish "good" from "poor" estimates?

One possibility: minimize the Sum of Squared Errors  
SSE

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

In a general sense, for a sample

and a model  $M$ ,

$$SSE = \sum_{i=1}^n (y_i - M(x_i))^2$$

## Linear regression: estimation

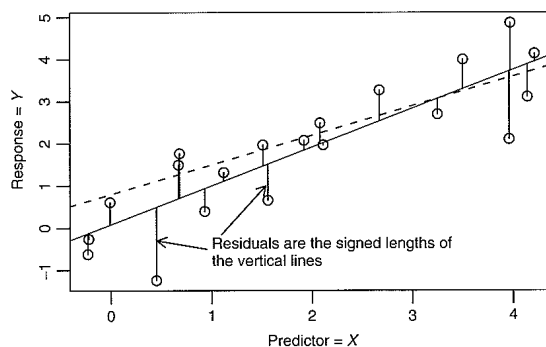
For a linear model, estimated parameters  $a, b$

$$SSE = \sum_{i=1}^n (y_i - a - b(x_i))^2$$

Estimation: choose parameters  $a, b$  so that the SSE is as small as possible. We call these: *least squares estimates*.

This *method of least squares* has an analytic solution for the linear case.

## Linear regression: residuals



**FIG. 2.3** A schematic plot for OLS fitting. Each data point is indicated by a small circle, and the solid line is a candidate OLS line given by a particular choice of slope and intercept. The solid vertical lines between the points and the solid line are the residuals. Points below the line have negative residuals, while points above the line have positive residuals.

## Linear regression: quality control

### Two parts:

Is the model adequate? (Residuals)

Are the parameter estimates good? (Confidence limits)

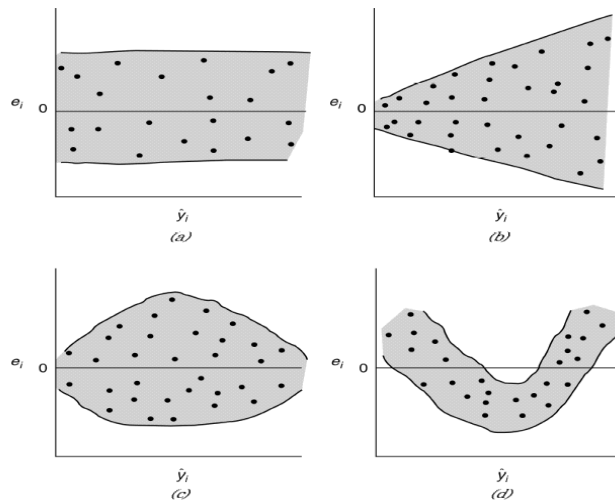
## Linear regression: quality control

Residual plots allow us to validate underlying assumptions:

- Relationship between response and regressor should be **linear** (at least approximately).
- Error term,  $\varepsilon$  should have zero mean
- Error term,  $\varepsilon$  should have **constant variance**
- Errors should be **normally distributed** (required for tests and intervals)

## Linear regression: quality control

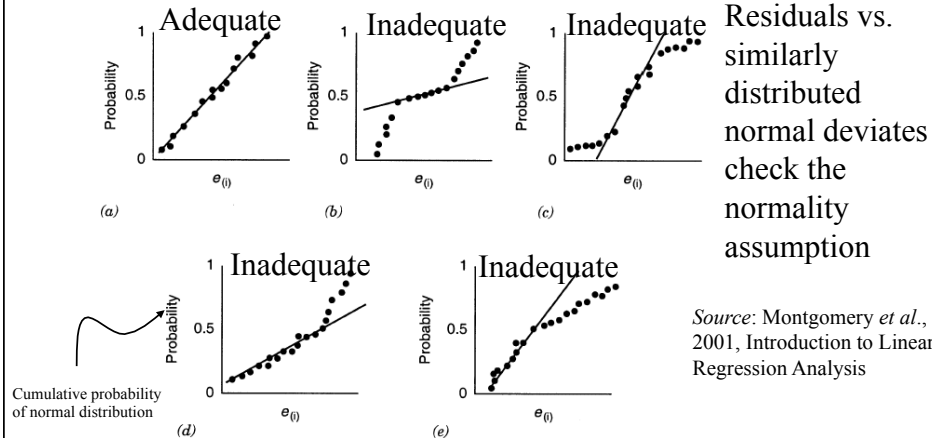
Source: Montgomery *et al.*, 2001, Introduction to Linear Regression Analysis



Check constant variance and linearity, and look for potential outliers.

What does our synthetic data look like, regarding this aspect?

# Linear regression: Q-Q plot



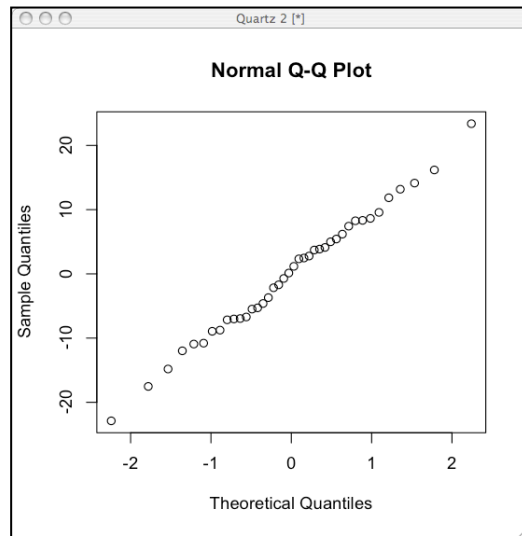
Residuals vs. similarly distributed normal deviates check the normality assumption

Source: Montgomery *et al.*, 2001, Introduction to Linear Regression Analysis

Figure 4.1 Normal probability plots: (a) ideal; (b) heavy-tailed distribution; (c) light-tailed distribution; (d) positive skew; (e) negative skew.

Q-Q plot: are the residuals normally distributed?

```
> qqnorm(res)
```



## Linear regression: Evaluating accuracy

If the model is valid, i.e. nothing terrible in the residuals, we can use it to predict. But how good is the prediction?

## Linear regression: Example in R

prediction and  
confidence limits

```
> pp<-predict(lm(dat[,2] ~ dat[,1]), int="p")
Warning message:
In predict.lm(lm(dat[,2] ~ dat[, 1]), int = "p") :
  Predictions on current data refer to _future_ responses

> pc<-predict(lm(dat[,2] ~ dat[,1]), int="c")

> pc
      fit      lwr      upr
1 60.99185 54.83484 67.14887
2 68.95974 64.90806 73.01143
3 79.41776 76.06824 82.76727
4 80.92002 77.36777 84.47226
5 62.14877 56.33094 67.96660
6 65.44705 60.54579 70.34832
7 69.31896 65.34347 73.29444
```

## Linear regression: Example in R

Plot limits

Sort on x

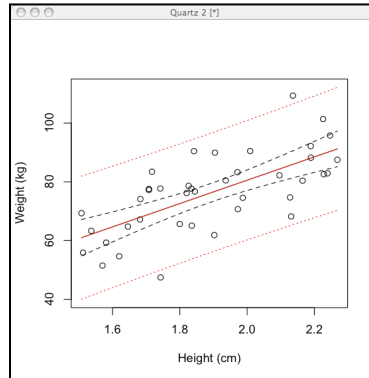
```
> o<-order(dat[,1])
> dat2<-dat[o,]
```

Recompute pp, pc

```
> pc<-predict(lm(dat2[,2] ~ dat2[,1]), int="c")
> pp<-predict(lm(dat2[,2] ~ dat2[,1]), int="p")
```

Plot

```
> plot(dat2, xlab="Height (cm)", ylab="Weight (kg)", ylim=range(dat2[,2], pp))
> matlines (dat2[,1], pc, lty=c(1,2,2), col="black")
> matlines (dat2[,1], pp, lty=c(1,3,3), col="red")
```

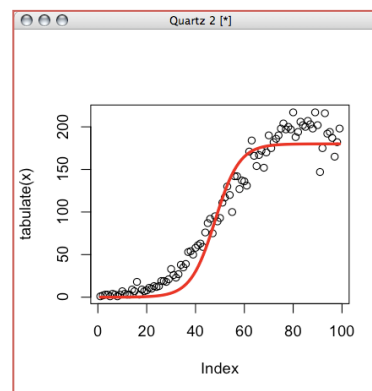


Module 7: Regression

bioinformatics.ca

## Nonlinear regression

- When you have a priori knowledge about the functional form of the model (from first principles);
- Uses *least squares fit* criterion for parameter estimation;
- Minimization in closed form (analytically) is usually not possible;
- Use numerical methods;
- Principle: calculate a *gradient* surface to guide the improvement of starting parameter choices.
- In R: **nls()**



Module 7: Regression

bioinformatics.ca

## Nonlinear regression

Nonlinear least square fitting in R is done with `nls()`.

```
res = nls(formula, data=data, start=c(parameters) )
```

In our example, the formula can be written:

```
> fz <- function(t, S, tm, B) { S*(1-(1/(1+exp(B*(t-tm)))))) }
```

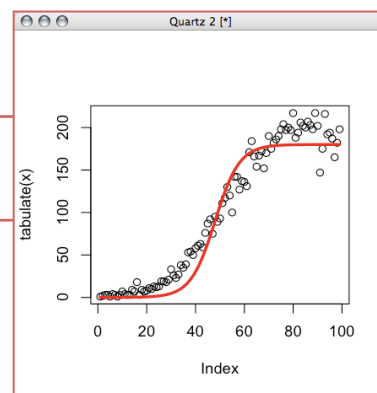
## Nonlinear regression

```
> fz <- function(t, S, tm, B) { S*(1-(1/(1+exp(B*(t-tm)))))) }
```

Try some reasonable starting parameters:

```
> curve(fz(x, S=180, tm=48, B=0.2),
        add=TRUE, col="red", lwd=3)
```

Good to go:



## Nonlinear regression

Invoke `nls()` with the formula and starting parameters:

```
> count <- tabulate(x)
> age <- c(1:99)
> res.fit <- nls(count ~ fz(age, S, med, B), start=c(S=180, med=48, B=0.2))
> res.fit
Nonlinear regression model
model: count ~ fz(age, S, med, B)
data: parent.frame()
      S      med      B
199.8084 49.5460 0.1020
residual sum-of-squares: 11854

Number of iterations to convergence: 7
Achieved convergence tolerance: 1.246e-06
```

```
> curve(fz(x, S=199.8084, tm=49.5460, B=0.102), add=TRUE,
        col="blue", lwd=3, lty=2)
```

## Nonlinear regression

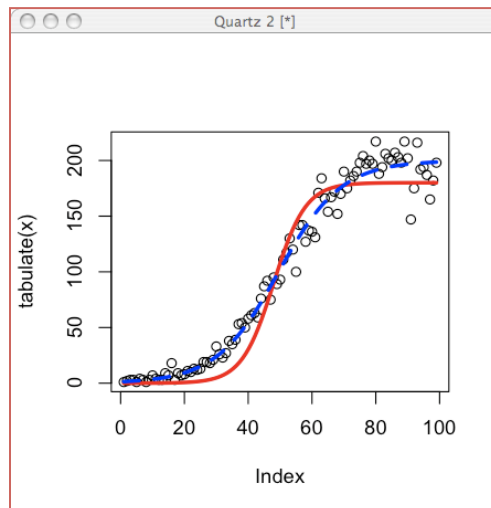
Invoke `nls()` with the formula and starting parameters:

```
> count <- tabulate(x)
> age <- c(1:99)
> res.fit <- nls(count ~ fz(age, S, med, B), start=c(S=180, med=48, B=0.2))
> res.fit
Nonlinear regression model
model: count ~ fz(age, S, med, B)
data: parent.frame()
      S      med      B
199.8084 49.5460 0.1020
residual sum-of-squares: 11854

Number of iterations to convergence: 7
Achieved convergence tolerance: 1.246e-06
```

```
> curve(fz(x, S=199.8084, tm=49.5460, B=0.102), add=TRUE,
        col="blue", lwd=3, lty=2)
```

## Nonlinear regression



Module 7: Regression

bioinformatics.ca

## Regression: summary

**Regression** is a *statistical technique* for investigating and **modeling** the relationship between variables, which allows:

- Parameter Estimation
- Hypothesis testing
- Use the Model (Prediction)

It's a powerful framework that can be readily generalized. You need to be familiar with your data, simulate it in various ways and check the model assumptions carefully!

Module 7: Regression

bioinformatics.ca

# Lab 7