Regression

Exploratory Data Analysis with **R**

Achilles and Ajax playing dice. Archaic (540-530 BCE)

Boris Steipe

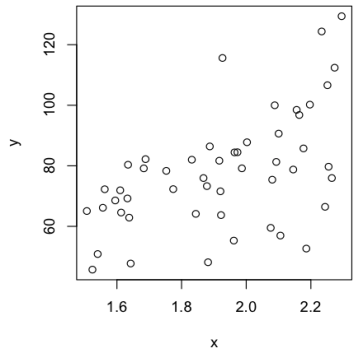
UNIVERSITY OF TORONTO

Department of Biochemistry
Department of Molecular Genetics

Learning Objectives

- Understand the effects of correlations on data;
- Understand linear and non-linear regression;
- Be able to compute linear regressions on data and evaluate them;
- Be aware of the (Maximum Information Coefficient) MIC as an alternative to the correlation coefficient for data mining.

Problem



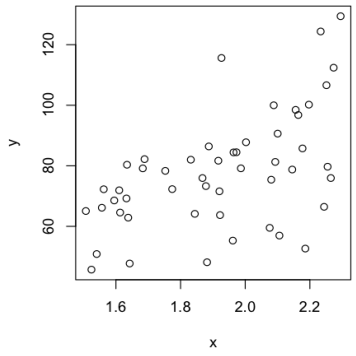
When we measure more one than one variable for each member of a population, a scatter plot may show us that the values are not completely independent: there is e.g. a trend for one variable to increase as the other increases.

Regression analyses the dependence.

Examples:

- Height vs. weight
- Gene dosage vs. expression level
- Survival analysis: probability of death vs. age

Correlation



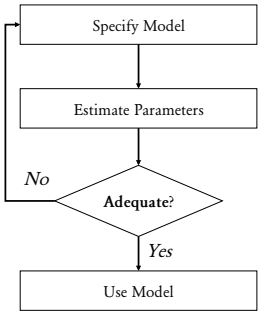
When one variable depends on the other, the variables are to some degree **correlated**.

(Note: correlation need not imply causality.)

In R, the function cov() measures covariance and cor() measures the Pearson coefficient of correlation (a normalized measure of covariance).

Pearson.s coeffecient of correlation values range from -1 to 1, with 0 indicating no correlation.

Modeling



Linear regression is one possible model we can apply to data analysis.

A model in the statistician.s sense might not be what you think ... it is merely a device to explain data. While it may help you consider mechanisms and causalities, it is not necessarily a representation of any particular physical or biological mechanism.

Note: correlation does not entail causation.

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."

ε_i are "errors" - not in the sense of being "wrong", but in the sense of creating deviations from the idealized model. The ε_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* β, and the *intercept* α.

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
    - ^{2} is constant, independent of x
  - ε_i are independent of each other
  - For proper statistical inference (CI, p-values), ε_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*

In a general sense, for a sample

and a model *M*,

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

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

Linear regression: estimation

For a linear model, with the estimated parameters *a, b*

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

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

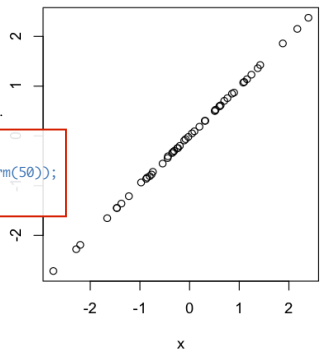
This *method of least squares* has an analytic solution for the linear case.

Pearson.s Coefficient of Correlation

How to interpret the correlation coefficient:

Explore varying degrees of randomness ...

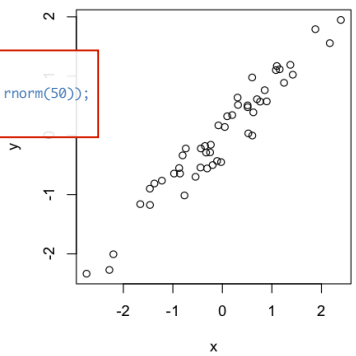
```
> x<-rnorm(50)
> r <- 0.99;
> y <- (r * x) + ((1-r) * rnorm(50));
> plot(x,y); cor(x,y)
[1] 0.9999666
```



Pearson.s Coefficient of Correlation

Varying degrees of randomness ...

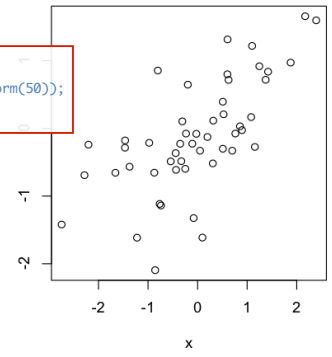
```
> x<-rnorm(50)
> r <- 0.8;
> y <- (r * x) + ((1-r) * rnorm(50));
> plot(x,y); cor(x,y)
[1] 0.9661111
```



Pearson.s Coefficient of Correlation

Varying degrees of randomness ...

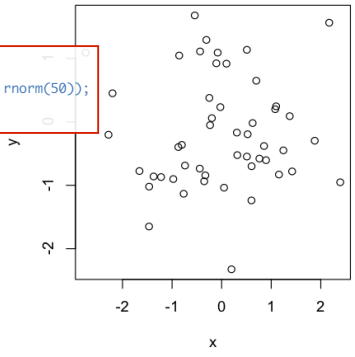
```
> x<-rnorm(50)
> r <- 0.4;
> y <- (r * x) + ((1-r) * rnorm(50));
> plot(x,y); cor(x,y)
[1] 0.6652423
```



Pearson.s Coefficient of Correlation

Varying degrees of randomness ...

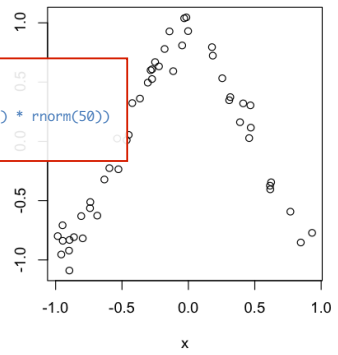
```
> x<-rnorm(50)
> r <- 0.01;
> y <- (r * x) + ((1-r) * rnorm(50));
> plot(x,y); cor(x,y)
[1] 0.01232522
```



Pearson.s Coefficient of Correlation

Non-linear relationships ...

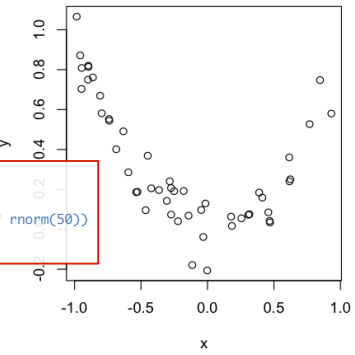
```
> x<-runif(50,-1,1)
> r <- 0.9
> # periodic ...
> y <- (r * cos(x*pi)) + ((1-r) * rnorm(50))
> plot(x,y); cor(x,y)
[1] 0.3438495
```



Pearson.s Coefficient of Correlation

Non-linear relationships ...

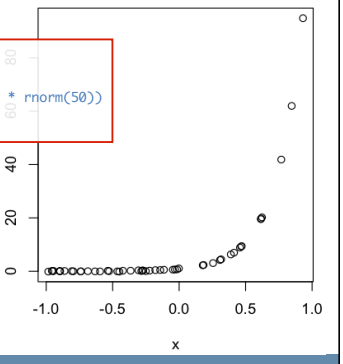
```
> x<-runif(50,-1,1)
> r <- 0.9
> # polynomial ...
> y <- (r * x*x) + ((1-r) * rnorm(50))
> plot(x,y); cor(x,y)
[1] -0.5024503
```



Pearson.s Coefficient of Correlation

Non-linear relationships ...

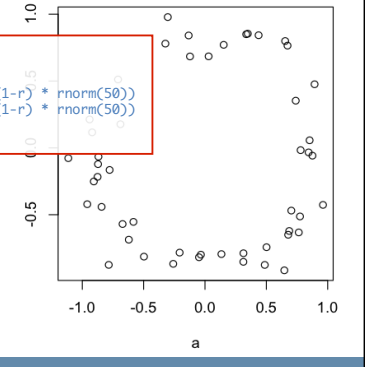
```
> x<-runif(50,-1,1)
> r <- 0.9
> # exponential
> y <- (r * exp(5*x)) + ((1-r) * rnorm(50))
> plot(x,y); cor(x,y)
[1] 0.6334732
```



Pearson.s Coefficient of Correlation

Non-linear relationships ...

```
> x<-runif(50,-1,1)
> r <- 0.9
> # circular ...
> a <- (r * cos(x*pi)) + ((1-r) * rnorm(50))
> b <- (r * sin(x*pi)) + ((1-r) * rnorm(50))
> plot(a,b); cor(a,b)
[1] 0.04531711
```



Linear regression: quality control

Intepreting the results has two parts:

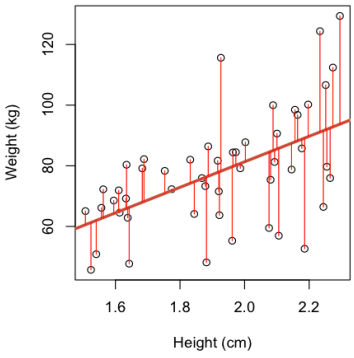
1: Is the model adequate? (Residuals)

2: Are the parameter estimates good? (Confidence limits)

Linear regression: residuals

Residuals:

The solid red line is the least-squares-fit line (regression line), defined by a particular slope and intercept. The red lines between the regression line and the actual data points are the residuals. Residuals are "signed" i.e. negative if an observation is smaller than the corresponding value of the regression line.



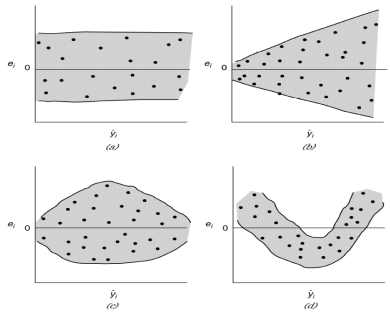
Linear regression: quality control

Residual plots allow us to validate underlying assumptions:

- Relationship between response and regressor should be **linear** (at least approximately).
- Error term, $\boldsymbol{\varepsilon}$ should have zero mean
- Error term, $\boldsymbol{\varepsilon}$ 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 example: height vs. weight

Get residuals:

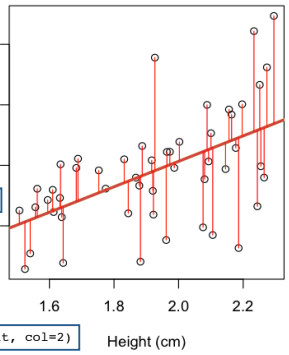
```
res <- resid(lm(HW[,2] ~ HW[,1]))
```

Get idealized values:

```
fit <- fitted(lm(HW[,2] ~ HW[,1]))
```

Plot differences:

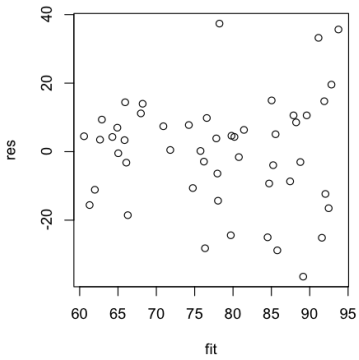
```
segments(HW[,1], HW[,2], HW[,1], fit, col=2)
```



Linear regression example: height vs. weight

fit vs. residuals

```
> plot(fit, res)
> cor(fit, res)
[1] -1.09228e-16
```



LINEAR REGRESSION: Q-Q PLOT

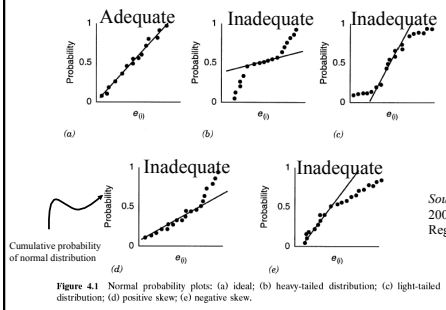


Figure 4.1 Normal probability plots: (a) ideal; (b) heavy-tailed distribution; (c) light-tailed distribution; (d) positive skew; (e) negative skew.

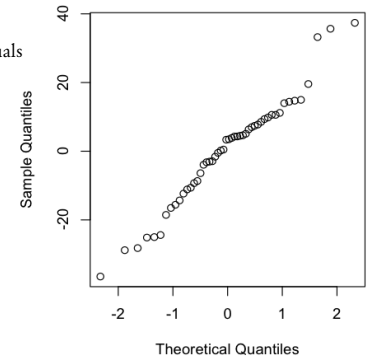
Residuals vs. similarly distributed normal deviates check the normality assumption

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

Linear regression example: height vs. weight

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: height vs. weight

prediction and confidence limits

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

> pc<-predict(lm(HW[,2] ~ HW[,1]), int="c")
> head(pc)
       fit      lwr      upr
1 60.57098 51.45048 69.69148
2 67.98277 61.53194 74.43360
3 77.96070 73.37784 82.54356
4 92.04435 84.23698 99.85171
5 76.34929 71.70340 80.99518
6 76.57656 71.94643 81.20670
```


Linear regression example: height vs. weight

Plot pp and pc limits

a: sort on x

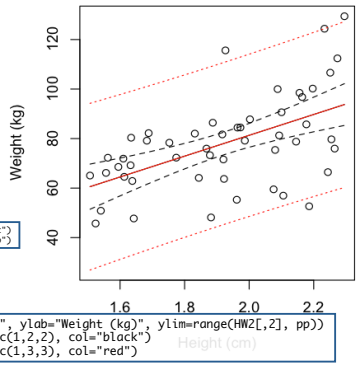
```
o <- order(HW[,1])
HW2 <- HW[o,]
```

b: recompute pp, pc

```
pc<-predict(lm(HW2[,2] ~ HW2[,1]), int="c")
pp<-predict(lm(HW2[,2] ~ HW2[,1]), int="p")
```

c: plot

```
> plot(HW2, xlab="Height (cm)", ylab="Weight (kg)", ylim=range(HW2[,2], pp))
> matlines (HW2[,1], pc, lty=c(1,2,2), col="black")
> matlines (HW2[,1], pp, lty=c(1,3,3), col="red")
```

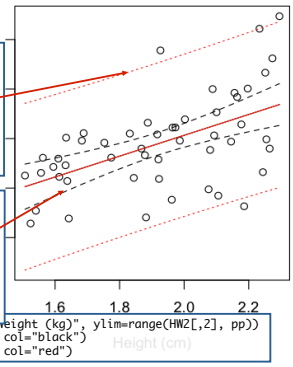


Linear regression example: height vs. weight

Plot pp and pc limits

prediction interval (at p=0.95) (values)

confidence interval (at p=0.95) (parameters)



Multiple regression

- Assume the variables act linearly against y
- Model: $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + e$
- Or $E(y \mid x) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$
- For two x-variables: $y = \alpha + \beta_1 x_1 + \beta_2 x_2$
- Estimated equations: $\hat{y} = a + b_1 x_1 + b_2 x_2$
- Least squares criterion for fitting the model (estimating the parameters)

$$SSE = \sum_{i=1}^{n} \left[y - (a + b_1 x_1 + b_2 x_2) \right]^2$$

- Solve for a, b_1, b_2 to minimize SSE

Nonlinear regression

nls() has much the same extractor functions as lm(). See the documentation.

The logistic function is useful for modelling risk/effect data in scenarios that have a binomial outcome: dead/alive, infected/healthy, cancer/cancer free *etc.*

It is especially useful to model the cumulative effect of independent risks. Assume a set of risk factors that each contribute to z

$$f(x) = \frac{1}{1 + Ae^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_k x_k}}$$

Nonlinear regression

$$y = \frac{1}{1 + Ae^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_k x_k}}$$

Rearrange:

$$\frac{(1-y)}{Ay} = e^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_k x_k}$$

$$\log \frac{Ay}{(1-y)} = -\beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_k x_k$$

... which can be analyzed with linear regression:

NONLINEAR REGRESSION

Applicable if you have *a priori* knowledge about the functional form of the model (from first principles);

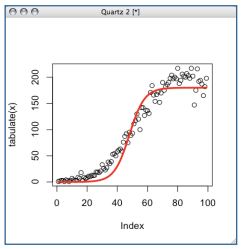
Uses *least squares fit* criterion for parameter estimation;

Minimization in closed form (analytically) is usually not possible;

Use numerical methods instead;

Principle: calculate a *gradient* surface to guide the improvement of starting parameter choices.

In R: **`nls()`**



Nonlinear regression: example

To define a non-linear model, let us consider a scenario in which a disease can be contracted with a probability that depends on the length of time a risk factor has been acting. This can be modelled as a "logistic function" - a type of function that was originally motivated by population growth considerations. Here N is a population, t is time and K is the fraction of population members who have contracted the disease.

$$N(t_{i+1}) - N(t_i) = rN(t_i)\left(1 - \frac{N(t_i)}{K}\right)$$

$$\frac{dN}{dt} = rN(t)\left(1 - \frac{N(t)}{K}\right)$$

Nonlinear regression

The logistic differential equation describes the change of a population if the change affects a fraction of the population at each interval of time.

$$\frac{dN}{dt} = rN(t)\left(1 - \frac{N(t)}{K}\right)$$

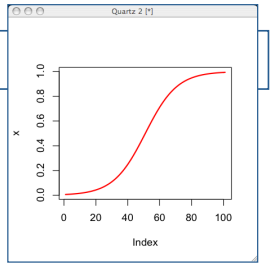
The "logistic function" is the solution to the logistic differential equation.

$$f(x) = \frac{C}{1 + Ae^{-Bx}}$$

Nonlinear regression: worked example

This is a generic simulation strategy - very powerful because it allows you to use an arbitrary function as a PDF (probability density function) and generate a corresponding set of samples.

```
x <- 1-(1/(1+exp(0.1*(-50:50))))
plot(x, type="l",
   col="red", lwd="2")
```

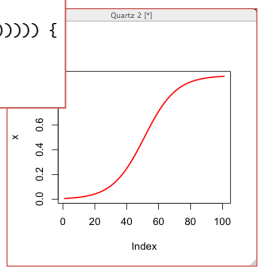


Nonlinear regression

Consider the function code:

```
age<-floor(runif(1,1,100))
s<-runif(1)
if (s < 1-(1/(1+exp(0.1*(age-50))))) {
  X <- append(X, age)
  i<-i+1
}
```

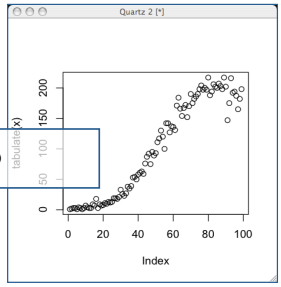
Two uniform random numbers are required:
1: select a candidate age
2: select a number between 0 and 1
If the number is larger than *f*(age), accept the choice, else, try a different age.



Nonlinear regression

We now have a simulated dataset with known parameters. How can we estimate (recover) the parameters?

```
1-(1/(1+exp(0.1*(age-50))))
```



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))))) }
```

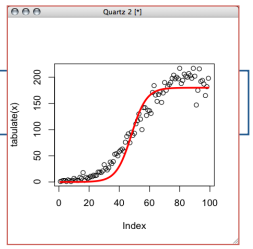

Nonlinear regression

```
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

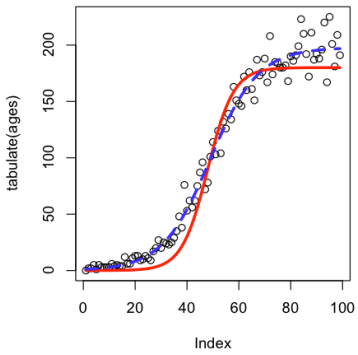
Invoke `nls()` with the formula and starting parameters:

```
> count <- tabulate(x)
> age <- c(1:99)
> nlsLogistic <- nls(count ~ fz(age, S, med, B), start=c(S=180, med=48, B=0.2))
> nlsLogistic
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
```

```
p <- coef(nlsLogistic)
curve(fz(x, S=p[1], tm=p[2], B=p[3]), add=TRUE, col="blue",
        lwd=3, lty=2)
```

Nonlinear regression



Regression: summary

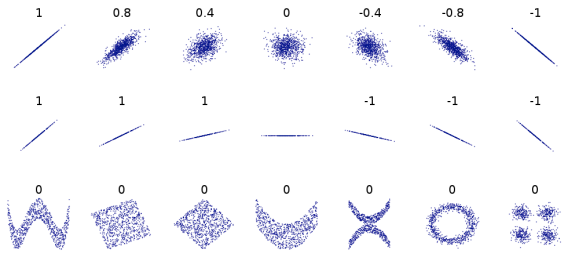
Regression is a *statistical technique* for investigating and **modeling** the relationship between variables, which allows:

- 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!

Correlation coefficient



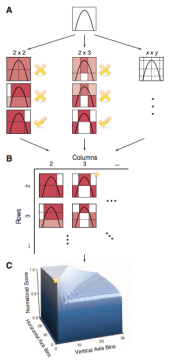
Maximal Information Coefficient

Detecting Novel Associations in Large Data Sets

Identifying interesting relationships between pairs of variables in large data sets is increasingly important. Here, we present a measure of dependence for two-variable relationships: the maximal information coefficient (MIC). MIC captures a wide range of associations both functional and not, and for functional relationships provides a score that roughly equals the coefficient of determination (R^2) of the data relative to the regression function. MIC belongs to a larger class of maximal information-based nonparametric exploration (MINE) statistics for identifying and classifying relationships. We apply MIC and MINE to data sets in global health, gene expression, major-league baseball, and the human gut microbiota and identify known and novel relationships.

Reshef *et al.* Science 2011, 334:1518

Maximal Information Coefficient



Computing MIC (A) For each pair (x,y), the MIC algorithm finds the x-by-y grid with the highest induced mutual information. (B) The algorithm normalizes the mutual information scores and compiles a matrix that stores, for each resolution, the best grid at that resolution and its normalized score. (C) The normalized scores form the characteristic matrix, which can be visualized as a surface; MIC corresponds to the highest point on this surface. In this example, there are many grids that achieve the highest score. The star in (B) marks a sample grid achieving this score, and the star in (C) marks that grid' s corresponding location on the surface.

Reshef *et al.* Science 2011, 334:1518

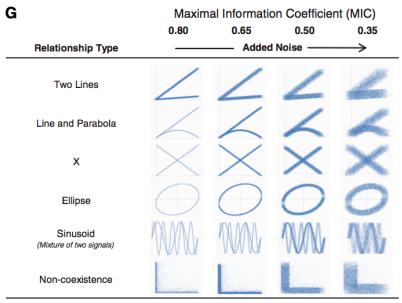
Maximal Information Coefficient

Relationship Type	MIC	Pearson	Spearman	Mutual Information (KDE)	Mutual Information (Kraskov)	CorGC (Principal Curve-Based)	Maximal Correlation
Random	0.18	-0.02	-0.02	0.01	0.03	0.19	0.01
Linear	1.00	1.00	1.00	5.03	3.89	1.00	1.00
Cubic	1.00	0.61	0.69	3.09	3.12	0.98	1.00
Exponential	1.00	0.70	1.00	2.09	3.62	0.94	1.00
Sinusoidal (Fourier frequency)	1.00	-0.09	-0.09	0.01	-0.11	0.36	0.64
Categorical	1.00	0.53	0.49	2.22	1.65	1.00	1.00
Periodic/Linear	1.00	0.33	0.31	0.69	0.45	0.49	0.91
Parabolic	1.00	-0.01	-0.01	3.33	3.15	1.00	1.00
Sinusoidal (non-Fourier frequency)	1.00	0.00	0.00	0.01	0.20	0.40	0.80
Sinusoidal (varying frequency)	1.00	-0.11	-0.11	0.02	0.06	0.38	0.76

Comparison of MIC to existing methods: Scores given to various noiseless functional relationships by several different statistics.Maximal scores in each column are accentuated.

Reshef *et al.* Science 2011, 334:1518

Maximal Information Coefficient



Performance of MIC on associations not well modeled by a function, as noise level varies.

Reshef *et al.* Science 2011, 334:1518

boris.steipe@utoronto.ca