tcdf - t cum dist fn

⇒ can do hyp. testing w/
ztest, ttest, ttest2
↑
2 samples

Ok, what if we want to test many samples? Suppose we look to see if there is a difference among many groups

We could do a t-test between each sample in a combinatoric fashion, but you are guaranteed a false positive! In text, talks about looking at relation between 8 groups ⇒ 7×8/2 = 28 combinations
Even random chance at 95% conf. level would lead to at least 1 false pos!

A better way is ANOVA, analysis of Variance

Basically, you look at the Variance of "supergroup" and Variance of all the subgroups.

If the variance of supergroups is larger than expected from subgroups, then at least one of the subgroups doesn't belong.

This ratio yields the F-statistic which has a dist. depending on the number of degrees of freedom.

Matlab has a canned routine anoval (A) which does this to a matrix of values.

Most of these formulas require independent normally distributed data.

If the data is not indep, $s_{\bar{x}}$ will be underestimated! (Effectively, n is reduced)

If the data isn't normally dist, sometimes you can fix it!

Example: Dist of particle size

- often have ∗ density of small particles >> larger ones
- could look at radius vs mass fraction, not ∗ of particles
- Also - log normal dist - work w/ log of variable rather than direct meas
- plot dist & see what works!

Linear Regression

In HW problem, wrote routine to fit a parabola to 3 points. In this case there was only one answer.

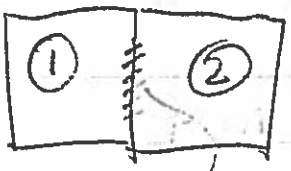
Often in data analysis you want to fit a curve to many data points

How do we do this? Use (usually) linear regression

This does not mean that we fit data with a line, rather that we use an algorithm which reduces the problem to a set of linear equations!

Let's look at a simple problem: measure mass transfer coefficients

Suppose we have a membrane between two reservoirs



↳ membrane area = A

We start with some concentration in reservoir ① of C_0 and we assume that reservoir ② is maintained at a constant C_{eq}. If the volume of reservoir ① is V, then:

$$V \frac{\partial C}{\partial t} = -Ah(C - C_{eq})$$

where $C\big|_{t=0} = C_0$

We can solve this equation:

$$C = C_{eq} + (C_0 - C_{eq})\, e^{-\left(\frac{htA}{V}\right)}$$

We wish to determine h by measuring C as a function of time

In order for us to use linear regression, the model must be linear in the modelling parameter,

In this case, the modelling parameter is h. We can rewrite the eqn:

$$(C - C_{eq}) = (C_0 - C_{eq})\, e^{-\frac{hA}{V}t}$$

$$\ln(C - C_{eq}) = \ln(C_0 - C_{eq}) - \frac{hA}{V}t$$

So if we plot $\ln(C - C_{eq})$ vs $\frac{At}{V}$

we get a line w/ intercept of $\ln(C_0 - C_{eq})$ and a slope of h!

How do we get the best fit line?

We look at the deviation of the points from the line, and try to minimize this in some way.

Let's take:

$$y \equiv \ln(C - C_{eq}), \quad a \equiv \frac{Ah}{V},$$

$$b \equiv \ln(C_0 - C_{eq})$$

We want to fit a series of points (t_i, y_i) by the model $y = at + b$

We shall use the method of least squares. We form the sum:

$$\text{Sum} = \sum_{i=1}^{N} (y_i - (at_i + b))^2$$

which is the sum of the squared distance between data pt. and model in the y direction.

We pick a & b so that this is a minimum

We let:

$$\left.\begin{aligned} \frac{\partial \text{sum}}{\partial a} &= 0 \\ \frac{\partial \text{sum}}{\partial b} &= 0 \end{aligned}\right\} \text{2 eqns for } a, b!$$

$$\frac{\partial \text{sum}}{\partial a} = \sum_{i=1}^{N} -2(y_i - (at_i + b))t_i$$

$$= \sum_{i=1}^{N} -2y_i t_i + 2a\sum_{i=1}^{N} t_i^2 + 2b\sum_{i=1}^{N} t_i = 0$$

and:

$$\frac{\partial \text{sum}}{\partial b} = \sum_{i=1}^{N} -2\left(y_i - (at_i + b)\right)$$

$$= \sum_{i=1}^{N} -2y_i + 2a\sum_{i=1}^{N} t_i + 2Nb = 0$$

This is a pair of linear equations for a & b!

We have the solution:

$$a\overline{t^2} + b\bar{t} = \overline{yt} \qquad (\text{let } \bar{x} \equiv \frac{1}{N}\sum_{i=1}^{N} x_i)$$

$$a\bar{t} + b = \bar{y}$$

$$\therefore\ a = \frac{\overline{yt} - \bar{y}\,\bar{t}}{\overline{t^2} - \bar{t}^2}$$

and $b = \bar{y} - a\bar{t}$

which can be used to get the mass transfer coefficient!

This is not the only way to approach the problem. Let's look at it as a system of equations!

We are trying to satisfy:

$$y_1 = a t_1 + b$$

$$y_2 = a t_2 + b$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$y_N = a t_N + b$$

Let's write this in <u>matrix form</u>:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \approx \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_N & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix}$$

we shall use $\underset{\sim}{X}$ to be the vector containing the modelling parameters:

$$X_1 \equiv a \qquad X_2 \equiv b$$

If we let the matrix of modelling functions be $\underset{\approx}{A}$ (e.g. $A_{11} = t_1$, etc.)

then we want to satisfy:

$$\underset{\sim}{y} \simeq \underset{\approx}{A}\,\underset{\sim}{x}$$

Actually, we want to minimize the residual:

$$\underset{\sim}{r} = \underset{\sim}{y} - \underset{\approx}{A}\,\underset{\sim}{x}$$

Or, for least squares, adjust $\underset{\sim}{x}$ s.t.:

$$\|\underset{\sim}{r}\|_2 \equiv \|\underset{\sim}{y} - \underset{\approx}{A}\,\underset{\sim}{x}\|_2 \text{ is as small as possible}$$

↑ Euclidean norm

We want to pick $\underset{\sim}{x}$ s.t.

$$\underset{\sim}{r}^T\underset{\sim}{r} \equiv (\underset{\sim}{y} - \underset{\approx}{A}\,\underset{\sim}{x})^T(\underset{\sim}{y} - \underset{\approx}{A}\,\underset{\sim}{x})$$

is a minimum.

This is exactly equivalent to what we did before!

Let's multiply this out:

$$r^2 \equiv \underset{\sim}{y}^T\underset{\sim}{y} - \underset{\sim}{y}^T\underset{\approx}{A}\underset{\sim}{x} - \underset{\sim}{x}^T\underset{\approx}{A}^T\underset{\sim}{y} + \underset{\sim}{x}^T\underset{\approx}{A}^T\underset{\approx}{A}\underset{\sim}{x}$$

We take the gradient of this w.r.t. $\underset{\sim}{x}$ and set it equal to zero!

$$\underset{\sim}{\nabla}_{\underset{\sim}{x}} r^2 = 0 = -2\underset{\approx}{A}^T\underset{\sim}{y} + 2\underset{\approx}{A}^T\underset{\approx}{A}\underset{\sim}{x} = 0$$

Thus we have the problem:

$$\underset{\approx}{A}^T\underset{\approx}{A}\underset{\sim}{x} = \underset{\approx}{A}^T\underset{\sim}{y} \quad !$$

This gives us a set of $\underline{\underline{n}}$ eq'ns for the n unknowns in $\underset{\sim}{x}$!

These are known as the normal equations.