Lecture 2 – Feb. 2012

General linear model: theory of linear model & advanced applications in statistics

Dr C. Pernet

Uni. of Edinburgh

Overview

- Linear algebra 2: Projection matrices
- ANOVAs using projections
- Multivariate Regressions
- Linear time invariant model (fMRI)
- A word on generalized linear model

Linear Algreba

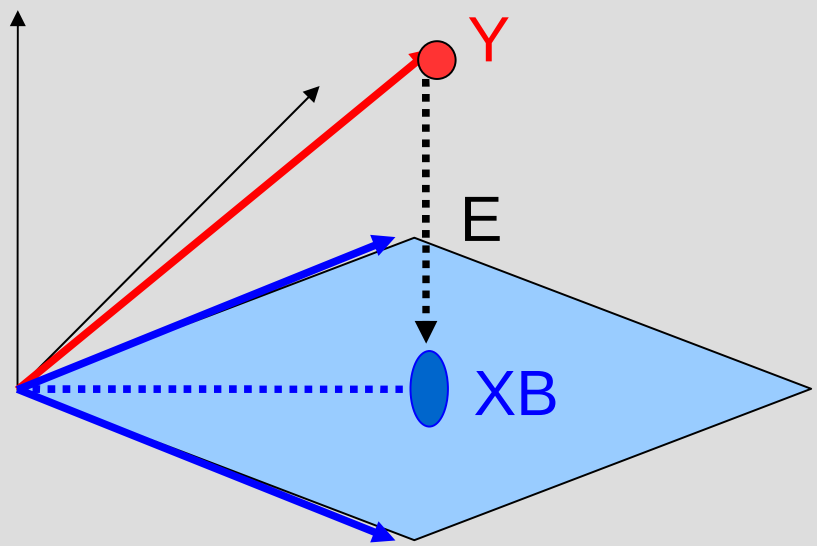
again!

Linear Algebra

- Lecture 1: we write equations as matrices – using matrices we can find all of the unknown (the Bs, or regression coefficients) in 1 single operation B = pinv(X)Y
- What we do with matrices, is to find how many of each vectors in X we need to be as close as possible to Y (Y = XB + e)

Linear Algebra and Statistics

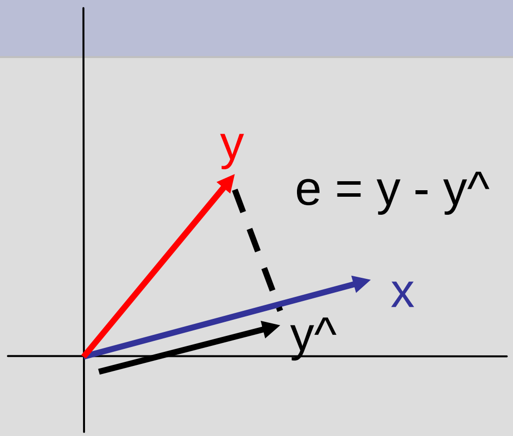
- Y = 3 observations X = 2 regressors
- Y = XB+E --> Y^=XB



SS total = variance in Y
SS effect = variance in XB
SS error = variance in E
R2 = SS effect / SS total
F = SS effect/df / SS error/dfe

We can find Y^ by computing B
Can we think of another way?

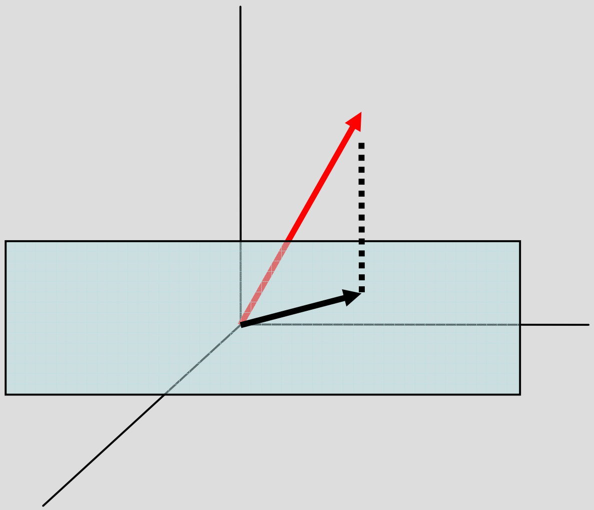
Linear Algebras: Projections



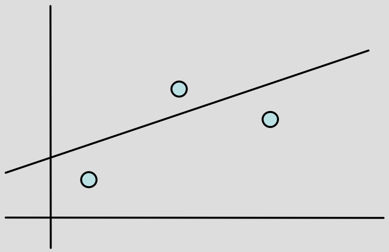
y^ = βx
x’(y- βx) = 0
βx’x = x’y
β = x’y / x’x

y^ = (x’y / x’x)x
y^ = Py → P = xx’ / x’x

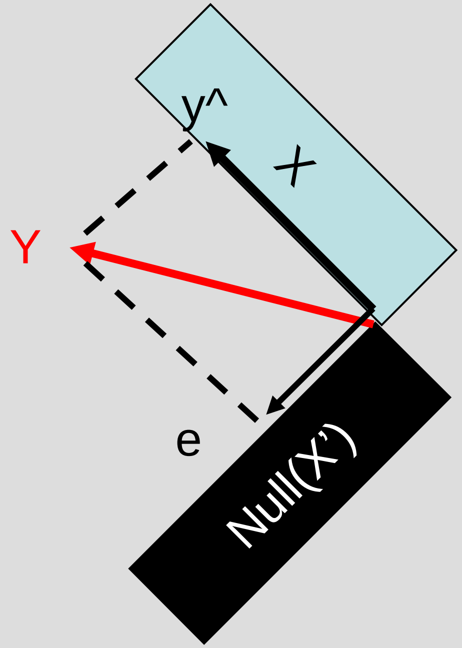
Why project? XB = Y may have no solution, the closest solution is a vector located in X space that is the closest to Y. With a bit of math we can find P = Xinv(X’X)X’



Projection and Least square



$y = \beta x + c$
P projects the points on the line
Minimizing the distance (^2)
is projecting at perpendicular angles



Y = y^+e
y^ = PY
e = (I-P)Y

ANOVA revised

1 way ANOVA

- u1 = rand(10,1) + 11.5; u2 = rand(10,1) + 7.2; u3 = rand(10,1) + 5; Y = [u1; u2; u3];
- x1 =[ones(10,1); zeros(20,1)]; x2 =[zeros(10,1); ones(10,1); zeros(10,1)]; x3 =[zeros(20,1); ones(10,1)]; X =[x1 x2 x3 ones(30,1)];
- <u>Lecture 1 solution:</u>
- B = pinv(X)*Y and Yhat = X*B
- <u>Now the solution:</u>
- P = X*pinv(X) and Yhat2 = P*Y

1 way ANOVA

- What to use? Both!
- Projections are great because we can now constrain Y^ to move along any combinations of the columns of X
- Say you now want to contrast gp1 vs gp2? C = [1 -1 0 0]
- Compute B so we have XB based on the full model X then using P(C(X)) we project Y^ onto the constrained model

1 way ANOVA

- R = eye(Y) – P; % projection on error space
- C = diag([1 -1 0 0]); % our contrast
- C0 = eye(size(X,2)) – C*pinv(C); % the opposite of C
- X0 = X*C0; % the opposite of C into X
- R0 = eye(size(Y,1)) – (X0*pinv(X0)); % projection on E
- M = R0 – R; % finally our projection matrix
- SSeffect = (Betas'*X'*M*X*Betas); % ~ 93.24
- F= (SSeffect / rank(X)-1) / (SSerror / size(Y,1)-rank(X)))

1 way ANOVA

- SStotal = norm(Y-mean(Y)).^2;
- SSerror = norm(Res-mean(Res)).^2;
- F = SSeffect / df(C) / SSerror / dfe(X)
- df = rank(C) – 1;
- dfe = length(Y) – rank(X);

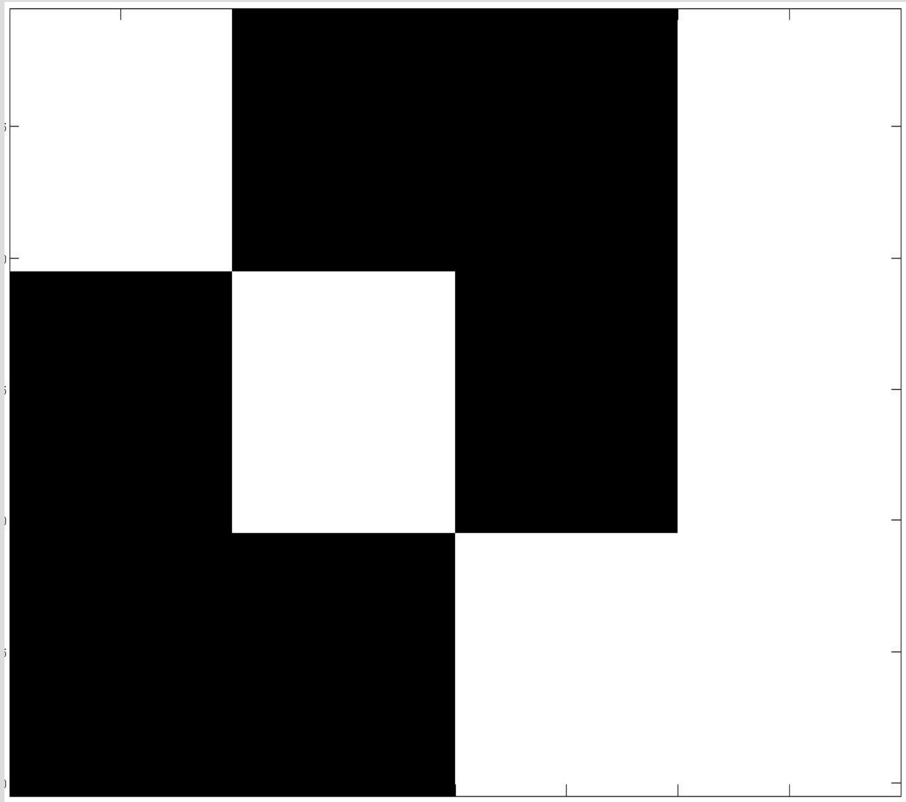
Code Summary

- Y = XB + e % master equation
- B = pinv(X)Y % betas
- P = Xpinv(X) % projection onto X
- R = I – P; % projection on null(X)
- SSerror = Y'RY; % makes sense
- C = diag(Constrast) → C0 → R0
- M = R – R0 % projection on C(X)
- SSeffect = B'X'MXB % our effect for C
- F = (SSeffect/rank(C)-1) /
  (SSerror/rank(X)-1);

Any ANOVAs

1 way

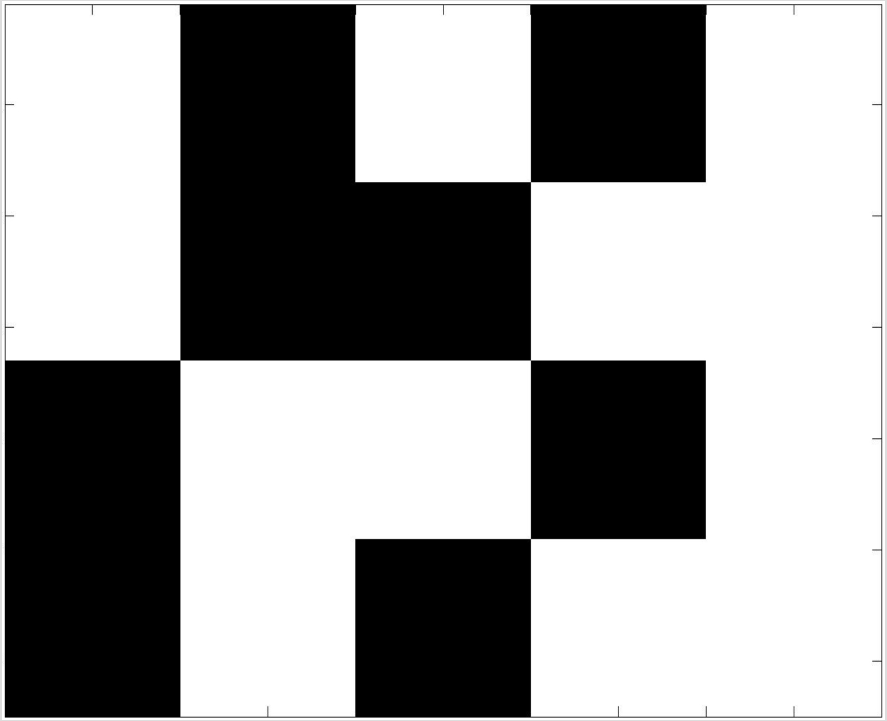
- F → C = diag([1 1 1 0])



How much the factor explain of the data (thus test against the mean)

N way

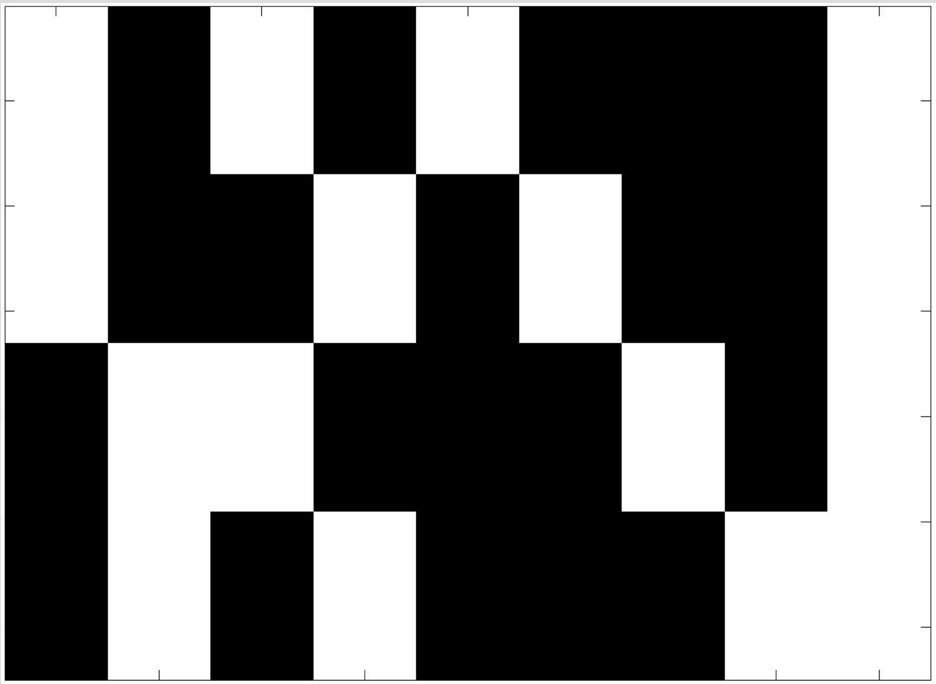
- F1 → C1 = diag([1 1 0 0 0])
- F2 → C2 = diag([0 0 1 1 0])



How much the factor i explain of the data (thus test against the other factors and the mean)

N way

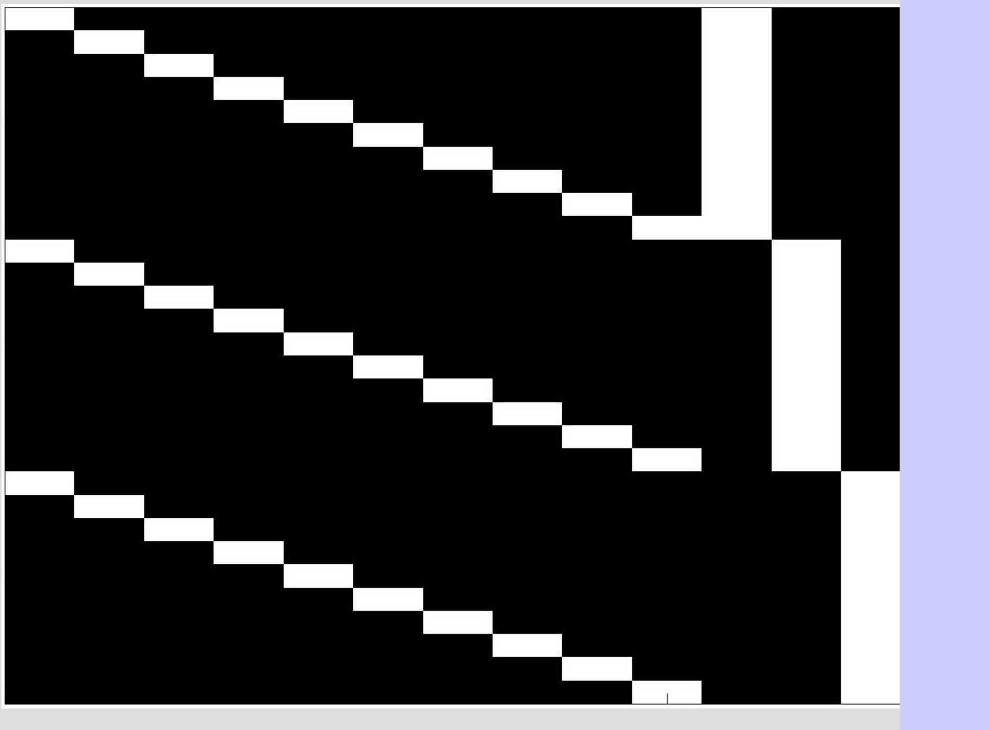
- Interaction: multiply columns
- → C = diag([0 0 0 0 1 1 1 1 0])



How much the interaction explains of the data (thus test against the main effects and the mean)

Repeated Measure

- S → Cs = diag([ones(10,1) 0 0 0 0])
- F → C = diag([zeros(10,1) 1 1 1 0])



The specificity of repeated measures is the subject effect

Note is this model SS error is the SS subject – there is no more grand mean, but a mean per subject

Multivariate Stats

Is this really more difficult?

Mutivariate stats

- Before we had one series of Y and a model X --> find B and various effects using C
- Now we have a set of Ys and a model X --> still want to find B and look at various effects
- IMPORTANT: the same model applies to all Ys

Multivariate regression

- We have one experimental conditions and plenty of measures – e.g. show emotional pictures and the subjects have to rate 1,2,3,4 – beside this subjective measurement the researcher measures RT, heart rate, pupil size
- Pblm: RT, heart rate, pupil size are likely to be correlated so doing 3 independent tests is not very informative.

Multivariate regression

- This can be solved easily since we apply the same model X to all the data Y
- Y = XB and B = pinv(X)Y
- B is now a matrix and the coef. for each Y are located on the diagonal of B
- SStotal, SSeffect and SSerror are also matrices called sum square (on the diagonal) and cross products matrices

Multivariate regression

- Load carbig
- Y = [Acceleration Displacement]
- X = [Cylinders Weights ones(length(Weigth),1)]

- Then apply the same technique as before
- Note the difference in results

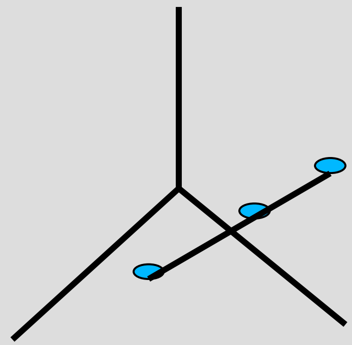
- Multivariate test depends on eigen values ~ PCA
- Take a matrix and find a set orthogonal vectors (eigen vectors) and weights (eigen values) such as $Ax = \lambda x$

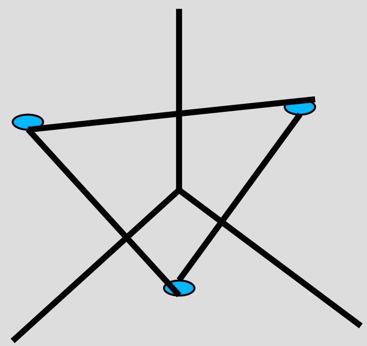
http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

Multivariate regression

- 4 tests in multivariate analyses, all relying on the eigen values λ of inv(E)*H

- Roy $\theta = \lambda 1 / 1 + \lambda 1$
- Wilk $\Lambda = \Pi(1/1+\lambda i)$
- Lawley-Hotelling $U = \Sigma \lambda i$
- Pillai $V = \Sigma(\lambda i/1 + \lambda i)$





Convolution model

Application to fMRI

General linear convolution model

- $y(t) = X(t)\beta + e(t)$
- $X(y) = u(t) \otimes h(\tau) = \int u(t-\tau)\, h(\tau)\, d\tau$

- The data y are expressed as a function of X which varies in time (X(t)) but β are time-invariant parameters (= linear time invariant model)

- X the design matrix describes the occurrence of neural events or stimuli u(t) convolved by a function h(τ) with τ the peristimulus time

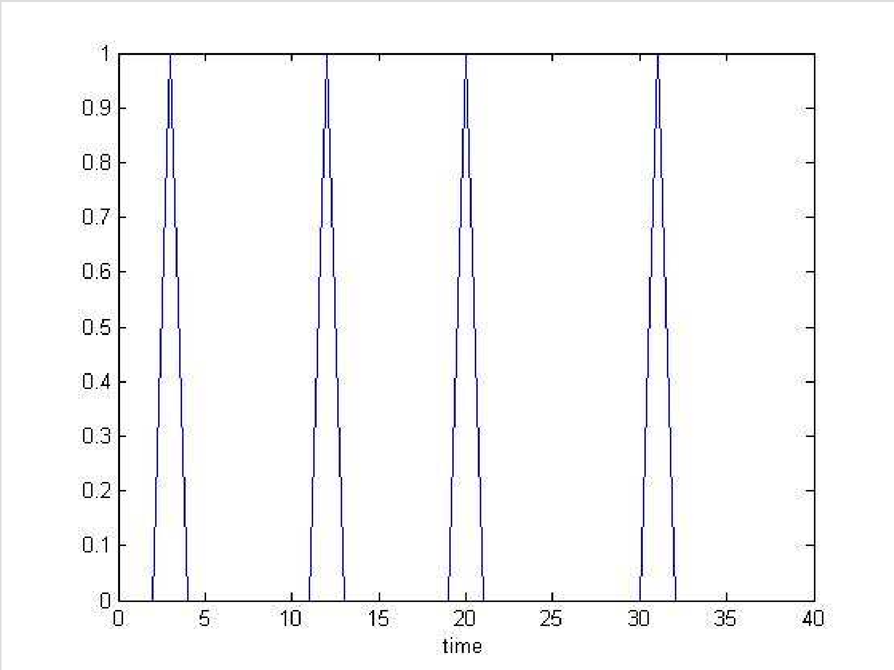
General linear convolution model

- **$X(y) = u(t) \otimes h(\tau)$**
- Say you have a stimulus (u) occurring every 8/12 sec and you measure the brain activity in between (y)
- If you have an a priori idea of how the brain response is (i.e. you have a function h which describes this response) then you can use this function instead of 1s and 0s

General linear convolution model

- **X(y) = u(t) ⊗ h(τ)**

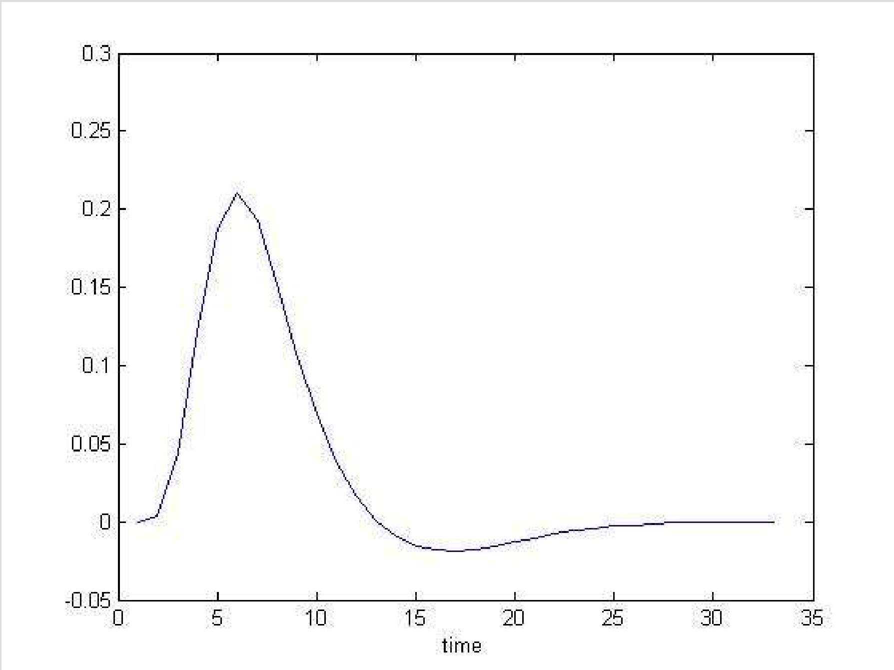
u = [0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0];



General linear convolution model

- **X(y) = u(t) ⊗ h(τ)**

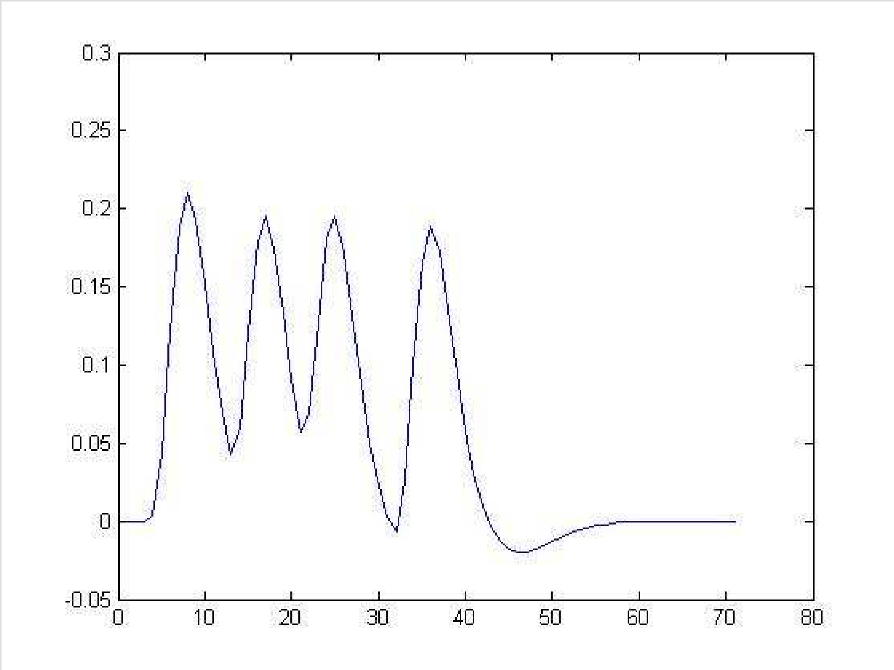
u = [0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0];
h = spm_hrf(1)



General linear convolution model

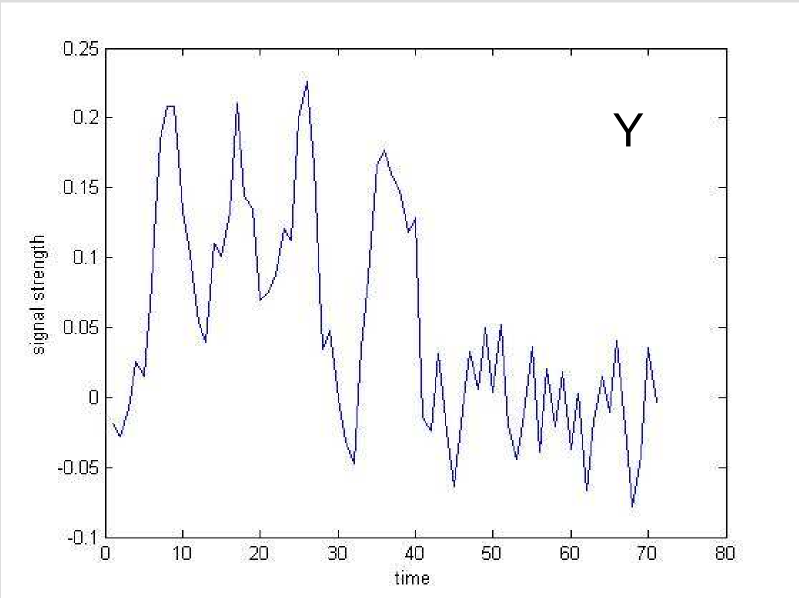
- **X(y) = u(t) ⊗ h(τ)**

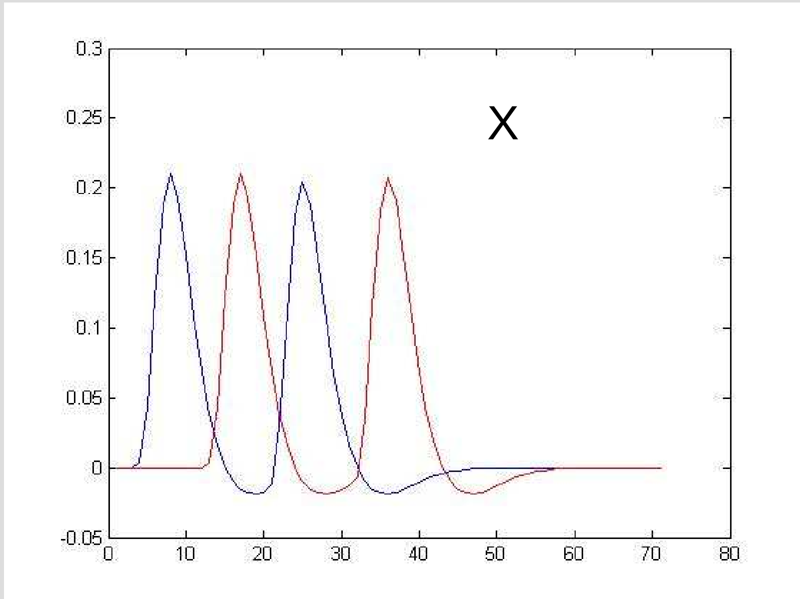
u = [0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0];
h = spm_hrf(1)
X = conv(u,h);



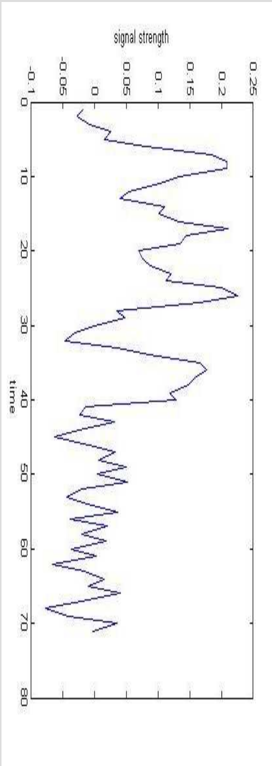
General linear convolution model

- y(t) = X(t)β + e(t)
- X has now two conditions u1 and u2 …
- And we search the beta parameters to fit Y

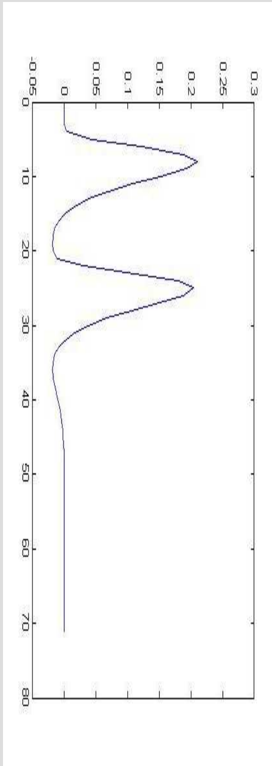




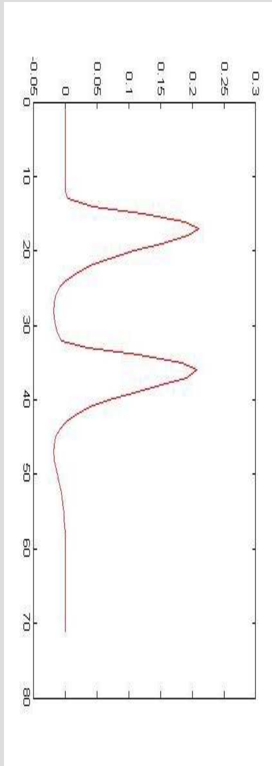
General linear convolution model



=



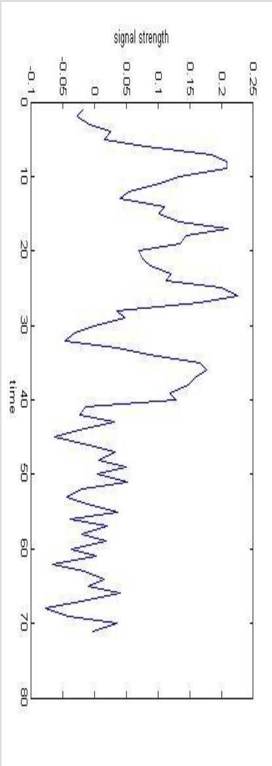
β1 +



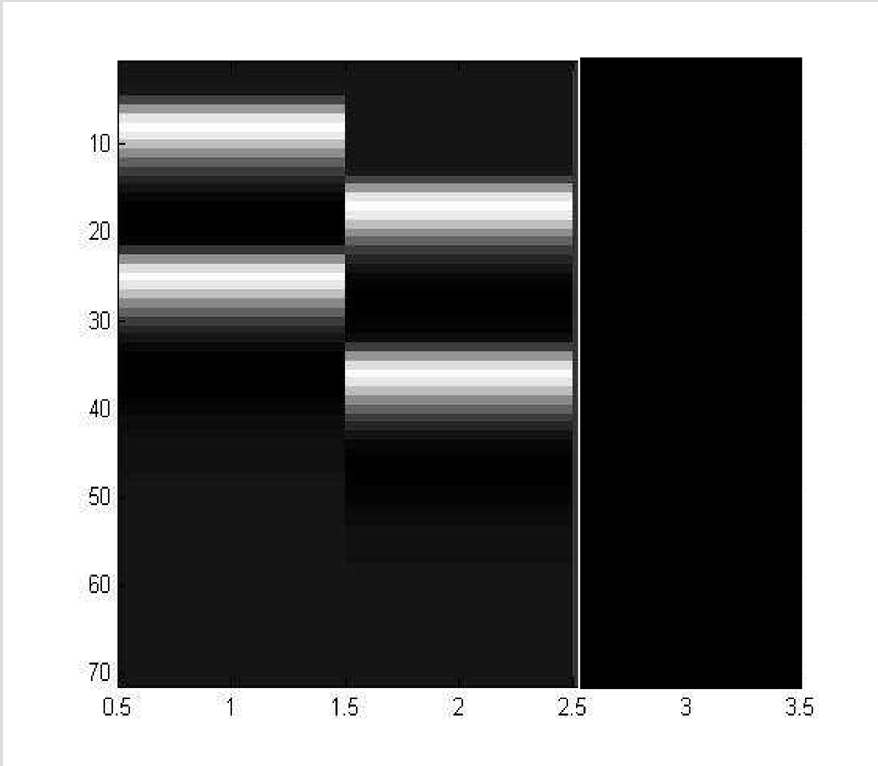
β2 + u + e

Data = cond. 1 * β1 + cond. 2 * β2 + cst + e

General linear convolution model



=



β1
β2 + e
u

fMRI (one voxel) = Design matrix * Betas + error

Generalized linear model

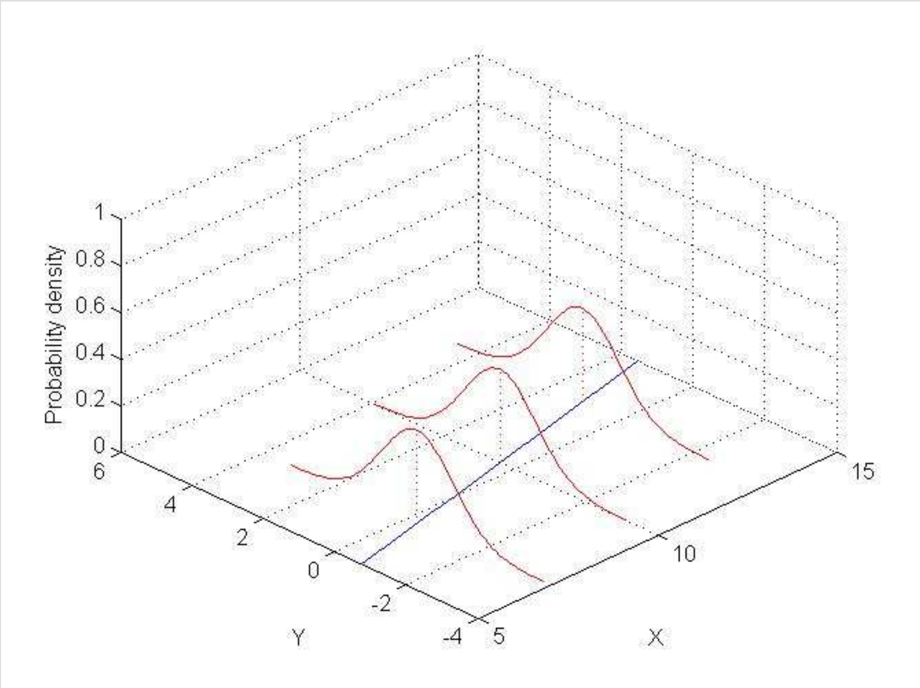
Move between
distributions

Generalized linear models

- You still have your responses Y but they follow a distribution that may be binomial, poisson, gamma, etc..
- You still make a model (design matrix) X and you search for a coefficient vector ß
- Here in addition, there is a link function f(.) such as f(Y)=Xß

Generalized linear models

- Usually, Y is a normally distributed response variable with a cste variance and for a simple case can be represented as a straight line (y = c + ax) with Gaussian distributions about each point

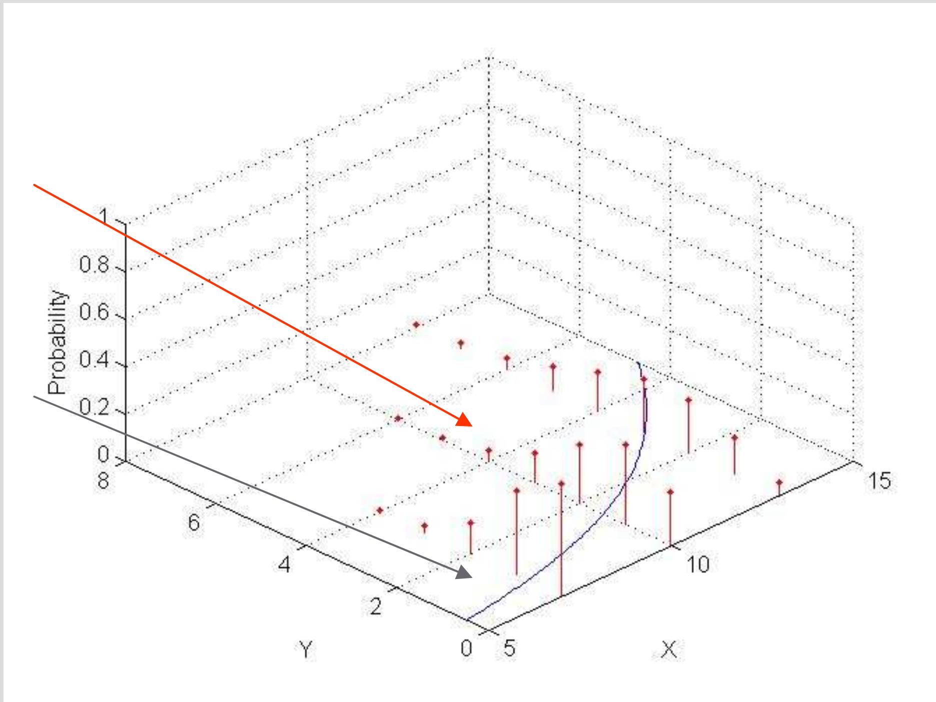


Generalized linear models

- In a generalized linear model, the mean of the response is modelled as a monotonic nonlinear transformation of a linear function of the predictors, g(b0 + b1*x1 + ...). The inverse of the transformation g is known as the "link" function.

Gamma distributed data

Link function

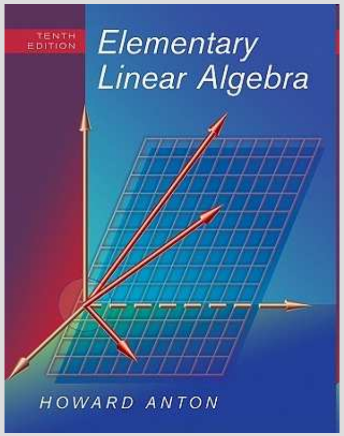


See glmfit

References

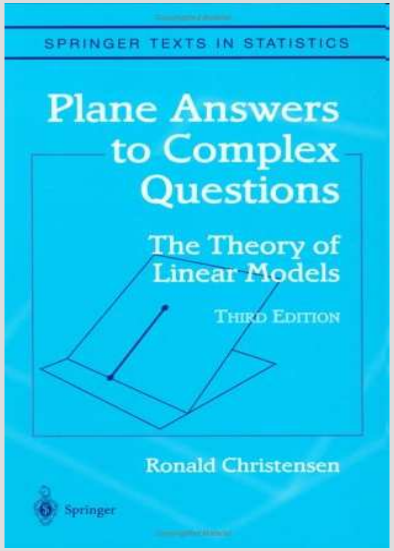
- **Linear algebra**
- MIT open course by prof. Strang (best course ever!)
- http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/

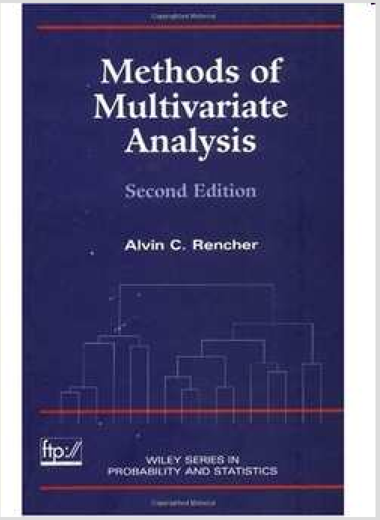
- Elementary Linear Algebra
By Howard Anton (best book ever! 10th edition)



References

- **<u>Stats with matrices</u>**
- Plane Answers to complex questions by Ronald Christensen





Methods of Multivariate Analysis by Alvin Rencher