General Linear Model

Cyril Pernet, PhD
SBIRC/SINAPSE – University of Edinburgh
http://www.sbirc.ed.ac.uk/cyril/cp_stats.html

Purpose of this lecture

- Give you the keys to understand how any statistics can be performed using the GLM

- What are the maths behind (easy)
- How is the GLM used in fMRI
- Key concepts (design matrix, linearity, independence, orthogonality, variance, contrasts)

Overview

- General **LINEAR** model: what is linear?
- General linear **MODEL**: what is a model?
- **GENERAL** linear model: why is that general?
- Even more general:

→ General linear **convolution** model (fMRI)

What is linearity?

Correlation analyses

What is a linear model?

- An equation or a set of equations that models data and which corresponds geometrically to straight lines, plan, cubes, hypercubes …

Linear models have two properties:

scaling: the magnitude of the system output is proportional to the system input ($y = ax$)

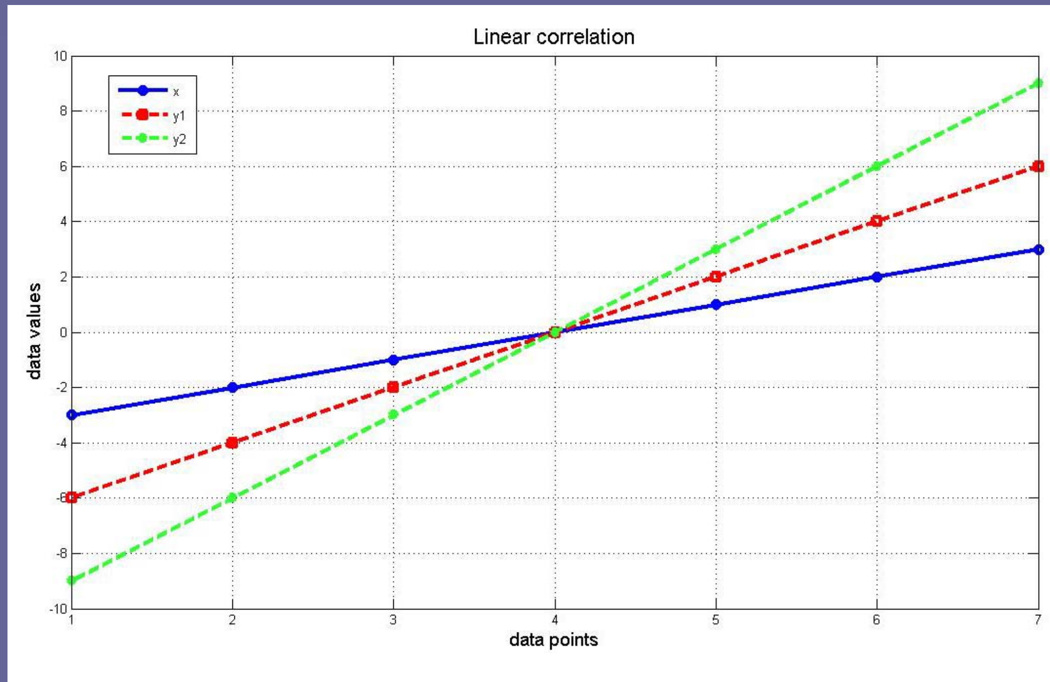
superposition: the total response to a set of inputs is the sum of individual inputs ($y = x1+x2$)

What is linear?

- ***Example of a linear correlation***

x = [-3 -2 -1 0 1 2 3];

<u>Scaling</u>:
y1 = 2*x;
<u>Superposition</u>
y2 = x+2*x;

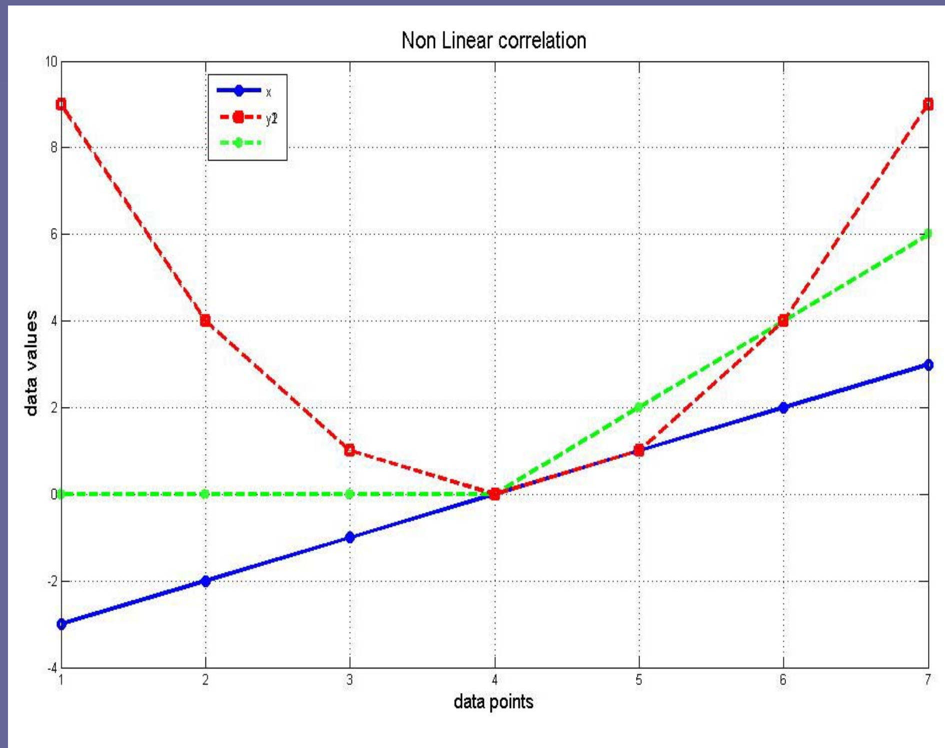


<u>Pearson corr. = 1</u>
corr([x' y'],'type','Pearson')

What is NOT linear?

- *Example of a non-linear correlation*

x = [-3 -2 -1 0 1 2 3];



<u>Non-linear scaling</u>
y1 = x.^2;
<u>Non-linear superposition</u>
y2 = x + abs (x)

Pearson corr. = 0
corr([x' y'],'type','Spearman') = 1 / 0.9

What is a linear model?

Regression analyses

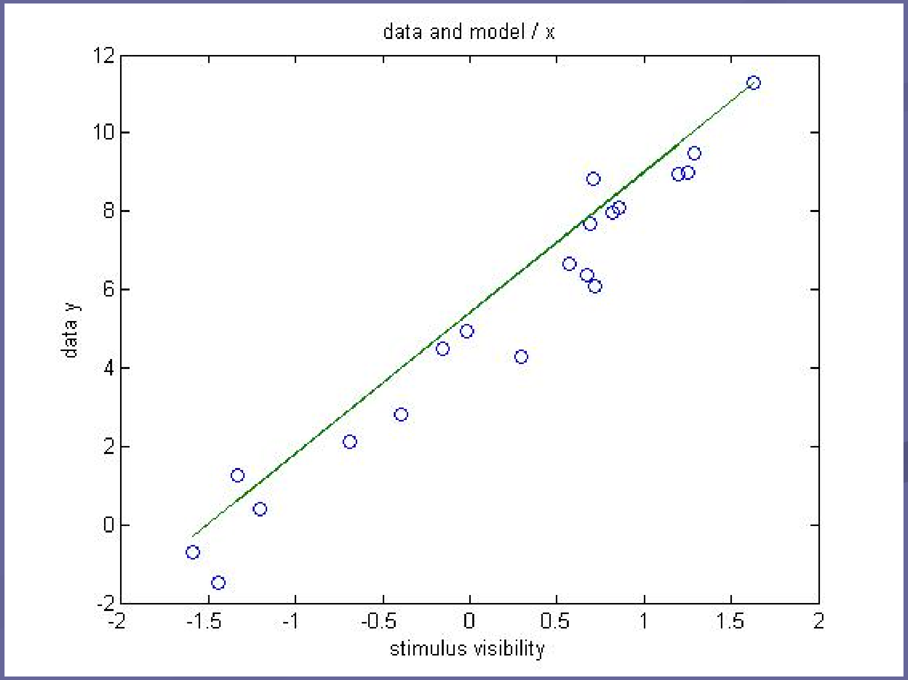
Simple regression

- An equation that models data and which corresponds geometrically a straight line

→ *simple regrression:* **y = βx + b (perf = β*age + b)**

x = randn(20,1);
y = 3*x + 5 + e

→ minimize the square distance between the line and each points = least squares fit



Multiple regression

- A equation that models data and which corresponds geometrically to a plan / cube ..

→ $y = \beta_1 x_1 + \beta_2 x_2 + b$ (perf = β_1*age + β_{2*}IQ+ b)

- It is again solved by the least squares method, i.e. one looks for coefficients (Betas) that minimizes the error, i.e. the difference between ($\beta_1 x_1 + \beta_2 x_2 + b$) and y
- This time, instead of a line, we will have a plan as we have 2 regressors

Digression into Linear Algebra

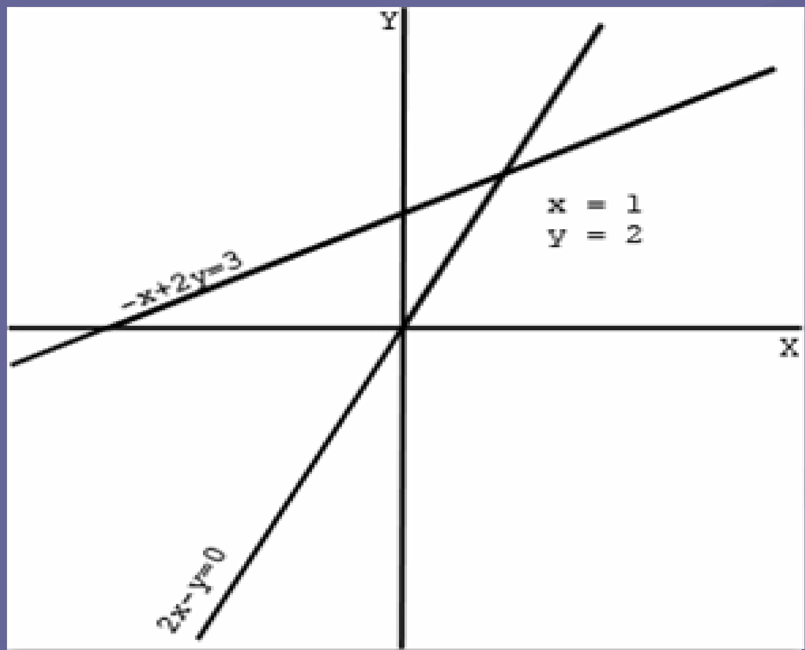
from equations to matrices

Linear equations

- Linear algebra has to do with solving linear systems, i.e. a set of linear equations
- For instance we have observations (y) for a stimulus characterized by its property x such as $\mathbf{y = \beta_1 x_1}$

$y = 2x$
$y = (3 + x) / 2$

<u>Solve</u>
$2x - y = 0$
$-x + 2y = 3$

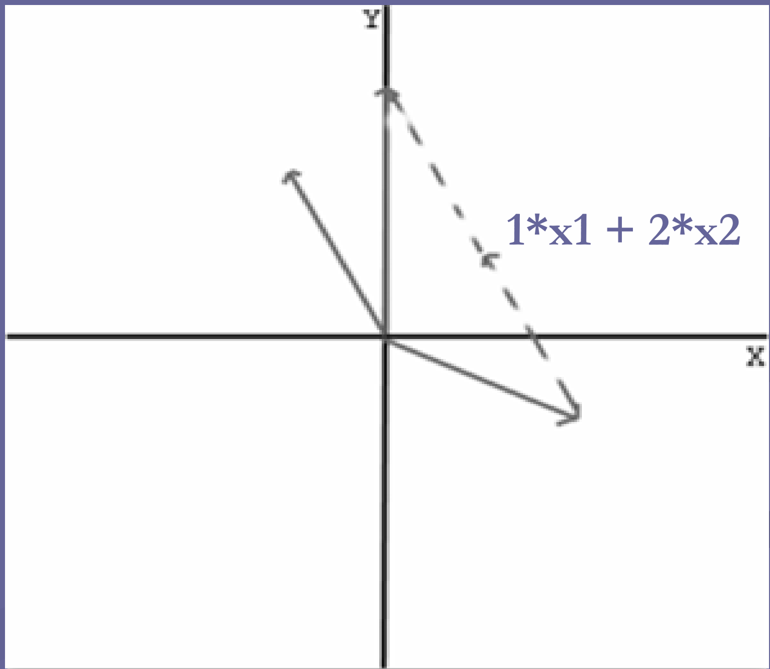


Linear equations

- With matrices, we change the perspective and try to combine columns instead of rows, i.e. we look for the coefficients with allow the linear combination of vectors

$$2x - y = 0$$
$$-x + 2y = 3$$

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$



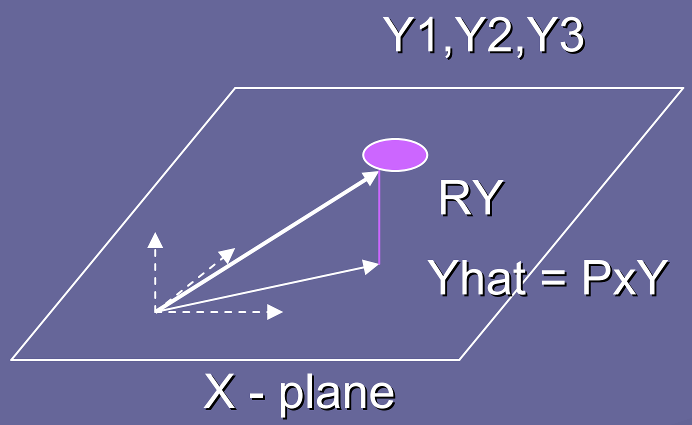
Linear equations

- The advantage of matrices is that one can handle a huge set a variables and numbers at once instead of working out the equations one by one ; in addition, it offers a fast and reliable way to solve a linear system

- With a single equation x β = y the solution is β = y(1/x)
- What about a square matrix X such as X β = y ?
- The solution is similar, the solution is Y * by the inverse of X (for non square design matrix X, solution is $inv(X^TX)X^TY$ or pinv(X)Y)

Geometrical perspective

- Y and Xβ are two points in $\mathbb{R}^J$ and we want to minimize the distance between those points.
- In addition the point Xβ lies in a subspace of $\mathbb{R}^J$ (dimension rank(X)) spanned by the columns of X
- The orthogonal projection of Y onto X is Yhat



$Y = X\beta + e$

$Px = X(X^TX)^{-1}X^T$

$e = RY$

$Yhat = X\beta$

Back to linear models (this time with matrices)

Regression analyses

Multiple regression

- Lets take an example

$$
\begin{bmatrix} 78.5000 \\ 74.3000 \\ 104.3000 \\ 87.6000 \\ 95.9000 \\ 109.2000 \\ 102.7000 \\ 72.5000 \\ 93.1000 \\ 115.9000 \\ 83.8000 \\ 113.3000 \\ 109.4000 \end{bmatrix}
=
\begin{bmatrix} c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \\ c \end{bmatrix}
+
\begin{bmatrix} 7\beta_1 \\ 1\beta_1 \\ 11\beta_1 \\ 11\beta_1 \\ 7\beta_1 \\ 11\beta_1 \\ 3\beta_1 \\ 1\beta_1 \\ 2\beta_1 \\ 21\beta_1 \\ 1\beta_1 \\ 11\beta_1 \\ 10\beta_1 \end{bmatrix}
+
\begin{bmatrix} 26\beta_2 \\ 29\beta_2 \\ 56\beta_2 \\ 31\beta_2 \\ 52\beta_2 \\ 55\beta_2 \\ 71\beta_2 \\ 31\beta_2 \\ 54\beta_2 \\ 47\beta_2 \\ 40\beta_2 \\ 66\beta_2 \\ 68\beta_2 \end{bmatrix}
+
\begin{bmatrix} 6\beta_3 \\ 15\beta_3 \\ 8\beta_3 \\ 8\beta_3 \\ 6\beta_3 \\ 9\beta_3 \\ 17\beta_3 \\ 22\beta_3 \\ 18\beta_3 \\ 4\beta_3 \\ 23\beta_3 \\ 9\beta_3 \\ 8\beta_3 \end{bmatrix}
+
\begin{bmatrix} 60\beta_4 \\ 52\beta_4 \\ 20\beta_4 \\ 47\beta_4 \\ 33\beta_4 \\ 22\beta_4 \\ 6\beta_4 \\ 44\beta_4 \\ 22\beta_4 \\ 26\beta_4 \\ 34\beta_4 \\ 12\beta_4 \\ 12\beta_4 \end{bmatrix}
+
\begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \\ e5 \\ e6 \\ e7 \\ e8 \\ e9 \\ e10 \\ e11 \\ e12 \\ e13 \end{bmatrix}
$$

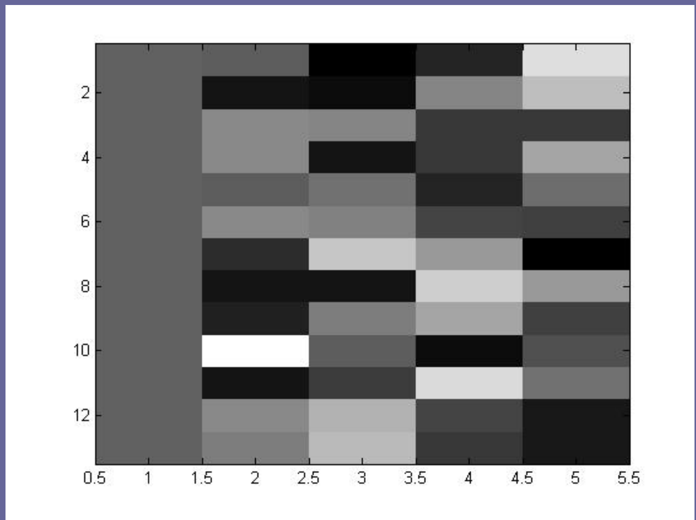
Multiple regression

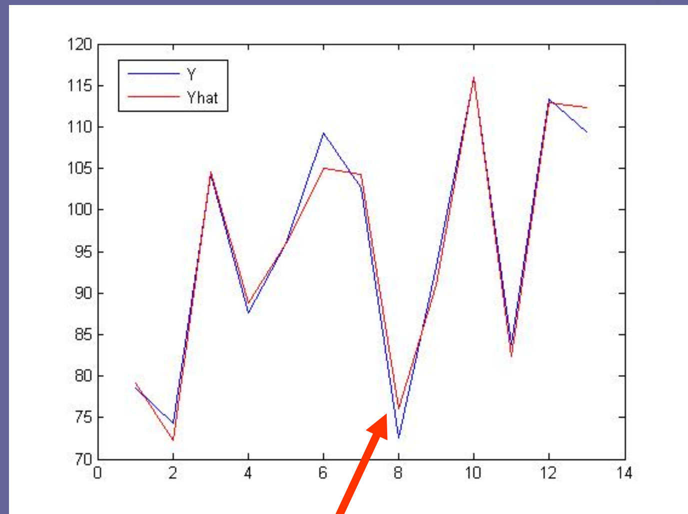
- In a Matrix form this can be rewritten as

$$
\begin{array}{ccccccc}
Y & = & X & & \beta & + & e \\
\begin{Bmatrix} 78.5000 \\ 74.3000 \\ 104.3000 \\ 87.6000 \\ 95.9000 \\ 109.2000 \\ 102.7000 \\ 72.5000 \\ 93.1000 \\ 115.9000 \\ 83.8000 \\ 113.3000 \\ 109.4000 \end{Bmatrix} & = &
\begin{Bmatrix} 1 & 7 & 26 & 6 & 60 \\ 1 & 1 & 29 & 15 & 52 \\ 1 & 11 & 56 & 8 & 20 \\ 1 & 11 & 31 & 8 & 47 \\ 1 & 7 & 52 & 6 & 33 \\ 1 & 11 & 55 & 9 & 22 \\ 1 & 3 & 71 & 17 & 6 \\ 1 & 1 & 31 & 22 & 44 \\ 1 & 2 & 54 & 18 & 22 \\ 1 & 21 & 47 & 4 & 26 \\ 1 & 1 & 40 & 23 & 34 \\ 1 & 11 & 66 & 9 & 12 \\ 1 & 10 & 68 & 8 & 12 \end{Bmatrix} & &
\begin{Bmatrix} c \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix} & + &
\begin{Bmatrix} e1 \\ e2 \\ e3 \\ e4 \\ e5 \\ e6 \\ e7 \\ e8 \\ e9 \\ e10 \\ e11 \\ e12 \\ e13 \end{Bmatrix} \\
\text{Data} & = & \text{Design matrix} & * & \text{Coef} & + & \text{Error}
\end{array}
$$

Multiple regression

- $Y = X\beta + e$ … we want $\beta = X^{-1}Y$
- The solution is $\beta = (X^TX)^{-1}X^TY \rightarrow$ X~Y (matlab inv, pinv)





The modelled data $Yhat = X\beta$

Multiple regression

- Stats:
- $Y = X\,\beta + e \rightarrow Yhat = X\,\beta \rightarrow e = Y\text{-}Yhat$

- R2 = SS effect / SS total

$$= \Sigma(\text{yhat i} - \text{mean(yhat)})^2 \;/\; \Sigma(\text{y i} - \text{mean(y)})^2$$

- F value = SS effect / SS error

$$= \frac{\Sigma(\text{yhat i} - \text{mean(yhat)})^2}{\text{rank(X)} - 1} \;/\; \frac{\Sigma(\text{e i} - \text{mean(e)})^2}{\text{N} - \text{rank(X)}}$$

Multiple regression

- <u>Usual problems in multiple regression</u>

→ Linear independence / colinearity

- **Colinearity**: indicates that a set of points are on a single straight line
- **Multicollinearity** is a statistical phenomenon in which two or more predictor variables are correlated
- **Linear independence:** in a family of vectors none can be describe as the linear combination of the others

Multiple regression

- Since we describe a regression model as a set of vectors, multicolinearity and linear independence have similar meaning

$$X = \begin{matrix} 1 & 3 & 5 \\ 3 & 8 & 14 \\ 4 & 5 & 13 \\ 8 & 6 & 22 \\ 7 & 8 & 22 \\ 4 & 1 & 9 \end{matrix}$$

→ x3 = 2x1+x2
→ corr([x1’, x3’])=.9
→ corr([x2’ x3’])=.7
→ rank(X) = 2

The matrix **rank** is the number of independent columns
(= effective df ; usually df = rank – 1 because of the cst term)

Intermediate summary

- Linear models are equations that describe lines, cubes, hypercubes, etc ..
- A regression is a linear model of the data
- It can be solve using linear algebra (matrices)

$$Y = X\beta + e$$

Data

Reg. coef.

Design matrix

Error

General Linear Model

All the stats with the same basic algebra

One-way ANOVA

GLM: 1 way ANOVA

Y	Gp
8	1
9	1
7	1
5	2
7	2
3	2
3	3
4	3
1	3
6	4
4	4
9	4

y(1..3)1= 1x1+0x2+0x3+0x4+c+e11
y(1..3)2= 0x1+1x2+0x3+0x4+c+e12
y(1..3)3= 0x1+0x2+1x3+0x4+c+e13
y(1..3)4= 0x1+0x2+0x3+1x4+c+e13

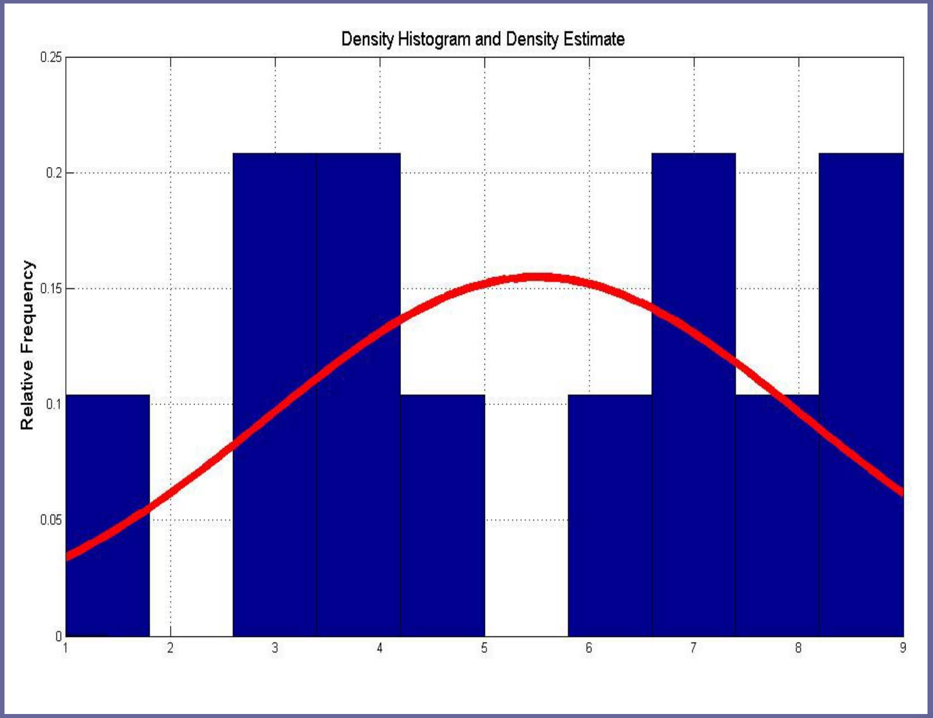
$$\begin{Bmatrix} 8 \\ 9 \\ 7 \\ 5 \\ 7 \\ 3 \\ 3 \\ 4 \\ 1 \\ 6 \\ 4 \\ 9 \end{Bmatrix} = \begin{Bmatrix} 1\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1 \\ 0\,1\,0\,0\,1 \\ 0\,1\,0\,0\,1 \\ 0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1 \end{Bmatrix} * \begin{Bmatrix} \beta 1 \\ \beta 2 \\ \beta 3 \\ \beta 4 \\ c \end{Bmatrix} + \begin{Bmatrix} e1 \\ \vdots \\ e13 \end{Bmatrix}$$

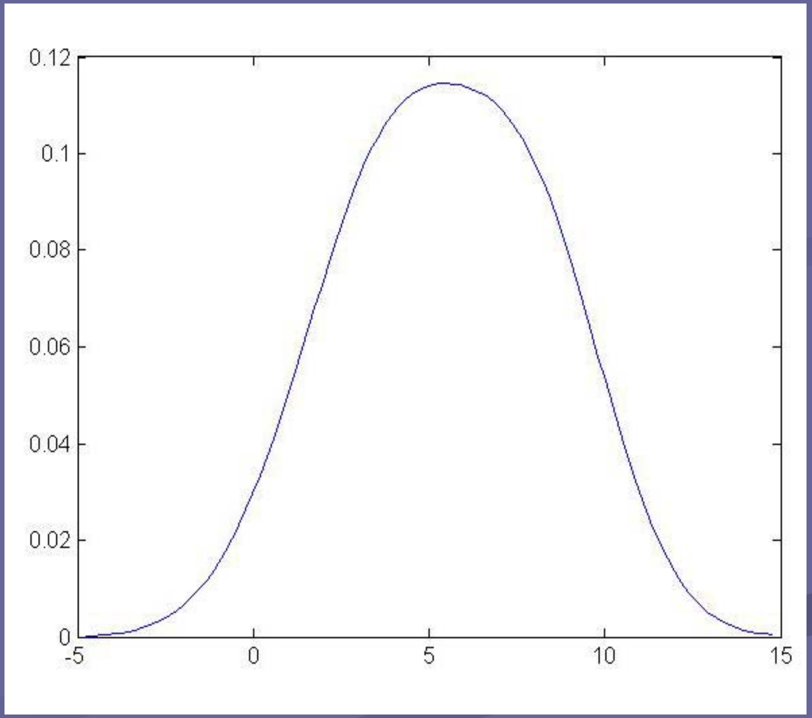
→ This is like the multiple regression except that we have ones and zeros instead of 'real' values

GLM: 1 way ANOVA

Y = [8 9 7 5 7 3 3 4 1 6 4 9]';

N = length(y); % = 12

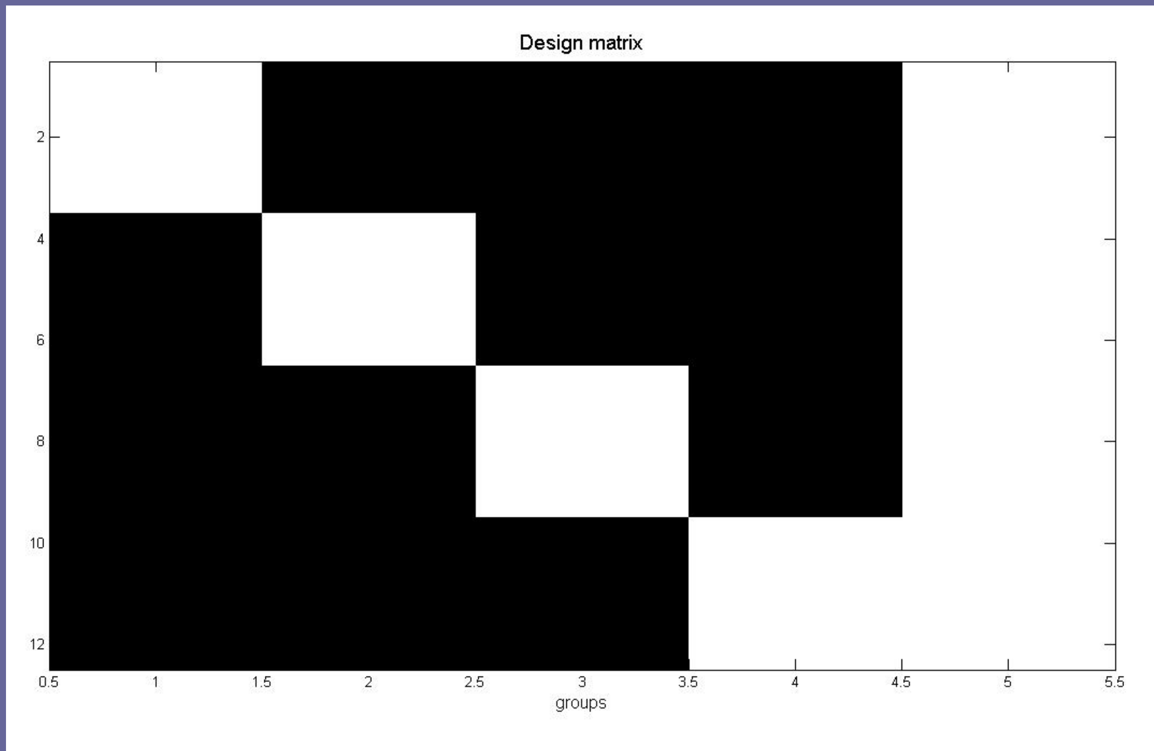




[f,z] = ksdensity(y); plot(z,f)

GLM: 1 way ANOVA

```
X = [kron(eye(4),ones(3,1)) ones(N,1)];
Imagesc(X); colormap(gray)
```

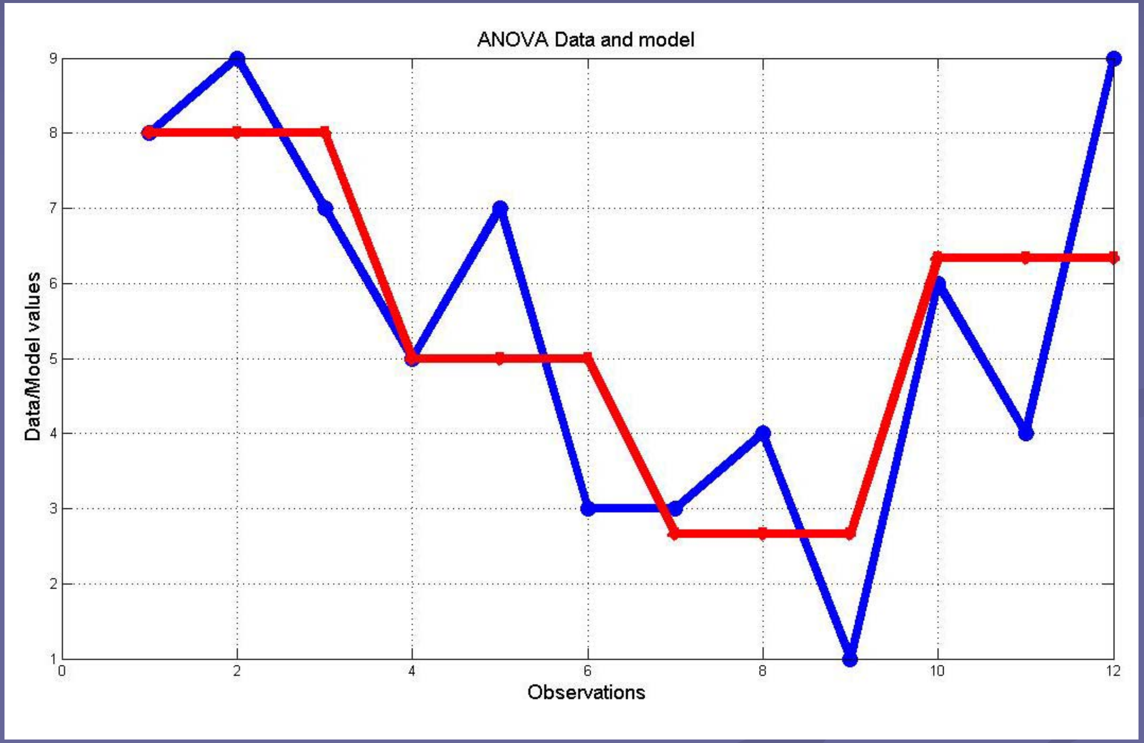


GLM: 1 way ANOVA

B = pinv(X)*Y;
Yhat = X*B;

8		8
9	Mean = 8	8
7		8
5	Mean = 5	5
7		5
3		5
3		2
4	Mean = 2	2
1		2
6		6
4	Mean = 6	6
9		6

Y → Yhat



GLM: 1 way ANOVA

SStotal = norm(y - mean(y)).^2 ; % another way to think about sum of squares is a squared distance in $\mathbb{R}^J$

SSeffect = norm(yhat – mean(yhat).^2 ; % same as above but in $\mathbb{R}^X$

SSerror = norm(e – mean(e)).^2; % this time in $\mathbb{R}^{X-1}$

R2 = SS effect / SS error

F = SS effect / rank(X)-1) / SS error / N-rank(x);

→ R square and F as for the regression ☺

GLM: contrasts

- Using a contrast C is like constraining the X space
- Then any test using a contrast matrix C will be like testing a combination of the columns again the error
  One can thus demonstrate that

$$F = \frac{\beta^T X^T M X \beta}{Y^T R Y} \cdot \frac{J-p}{p1}$$

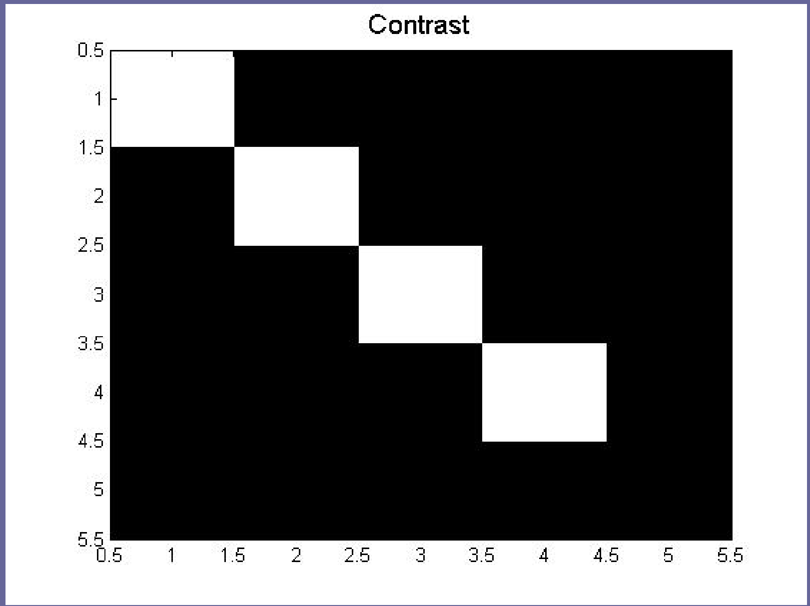
M = R0 – R
R0 = I – X0pinv(X0)
X0 = XC0
C0 = I – Cpinv(C)

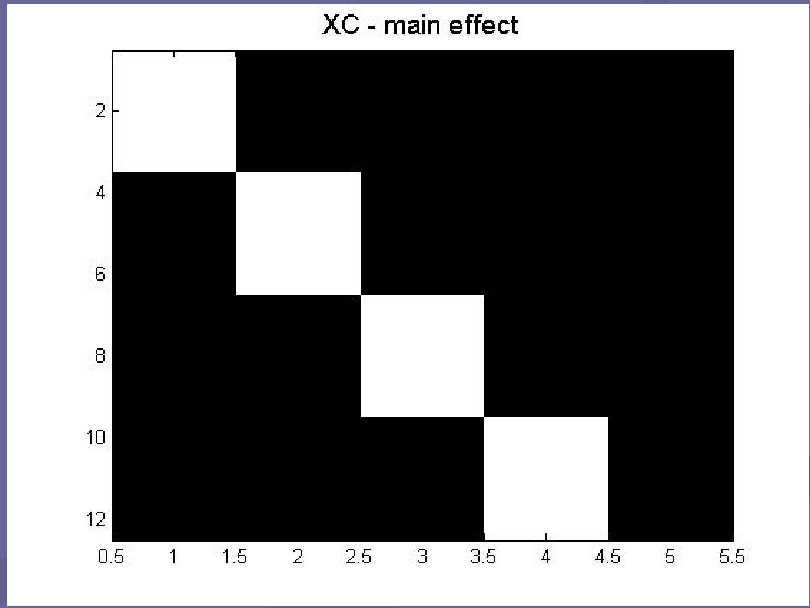
GLM: contrasts

C = eye(5); C(5,5) = 0;

% main effect = all columns except the grand avg

Xc = X*C; % this is the model to test



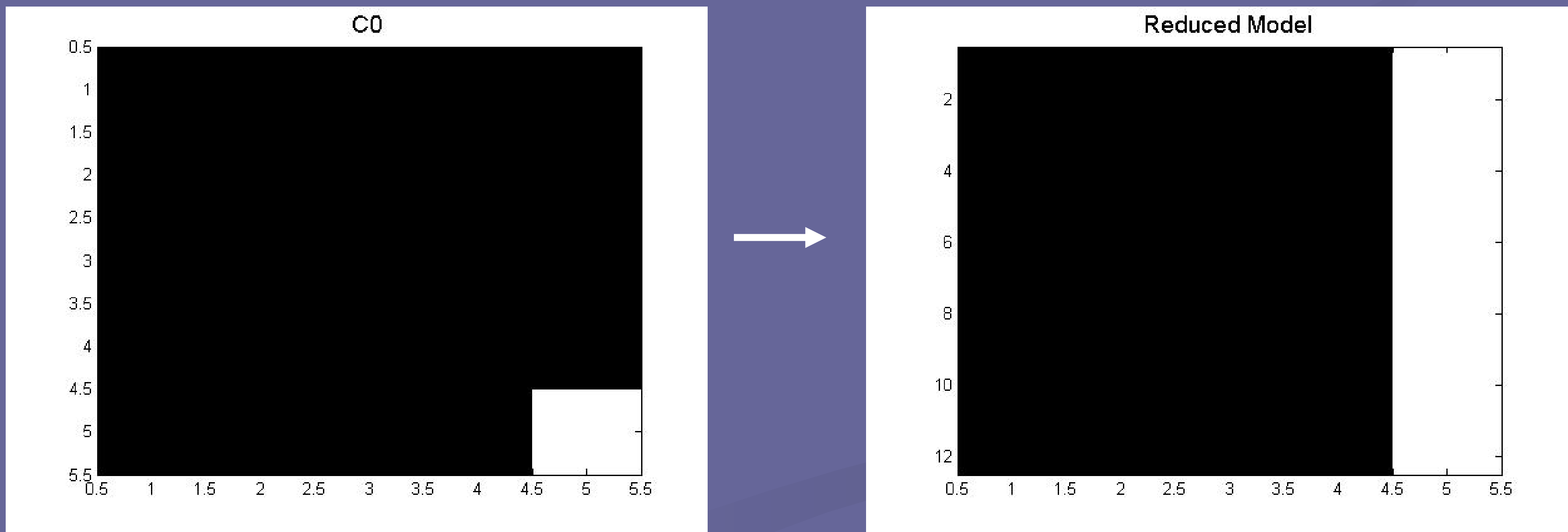


GLM: contrasts

C = eye(5); C(5,5) = 0; Xc = X*C

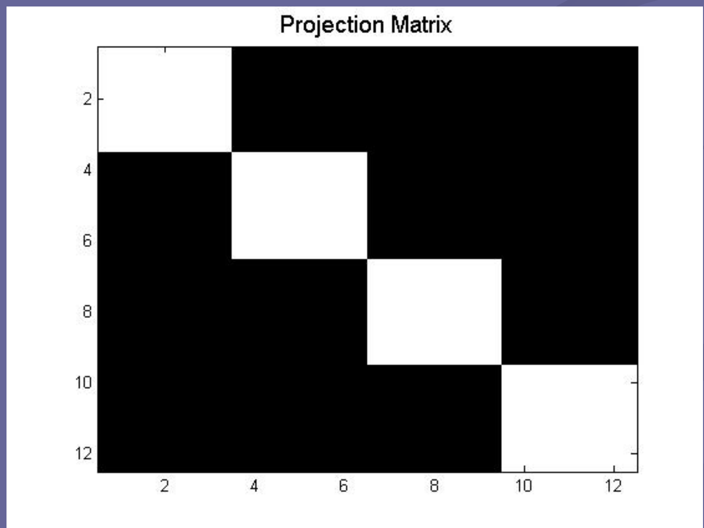
C0 = eye(rank(X)+1) - C*pinv(C); % this is the opposite of C

X0 = X*C0; % this is thus the reduced model



GLM: contrasts

```
C = eye(5); C(5,5) = 0; Xc = X*C;
C0 = eye(rank(X)+1) - C*pinv(C); X0 = X*C0;
R = eye(length(Y)) - (X*pinv(X)); % residual matrix for X
R0 = eye(length(Y)) - (X0*pinv(X0)); % residual matrix for X0
M = R0 - R; % M is the projection matrix onto Xc
```



GLM: contrasts

C = eye(5); C(5,5) = 0; Xc = X*C;

C0 = eye(rank(X)+1) - C*pinv(C); X0 = X*C0;

R = eye(length(Y)) - (X*pinv(X));

R0 = eye(length(Y)) - (X0*pinv(X0));

M = R0 - R; dfe = length(Y)-rank(X);

F = ((beta'*X'*M*X*beta)/(rank(C)-1))/((Y'*R*Y)/dfe);

= 4.45 ☺

df = rank(C) - 1

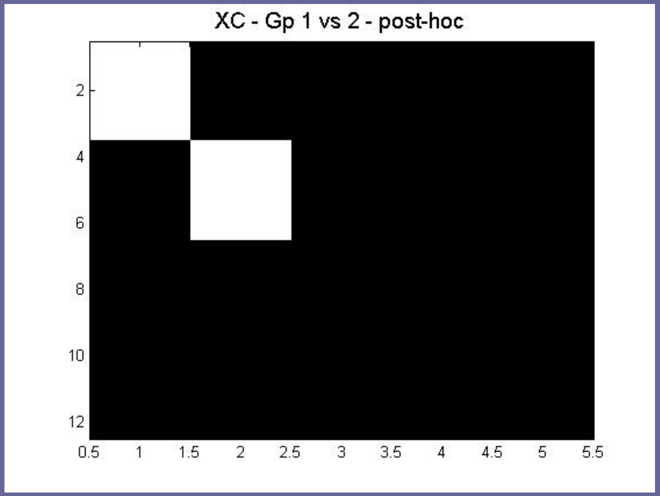
= Yhat'*Yhat projected onto Xc

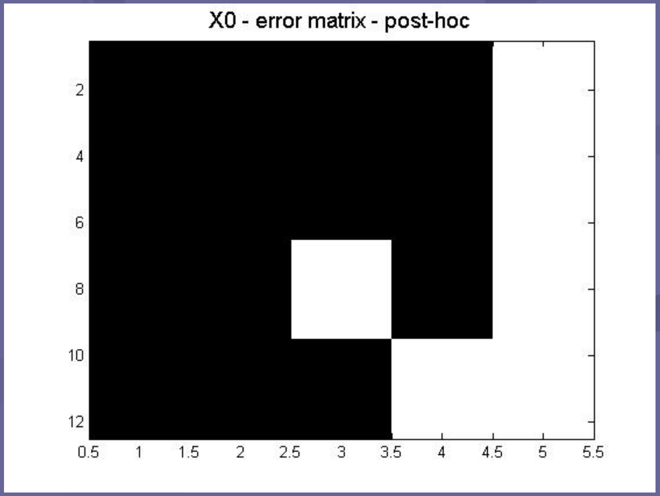
GLM: contrasts

- We can also contrast only 2 columns for a post-hoc test – note that the error matrix will contains all the other columns, i.e. ≠ t-test between these 2 columns only

C = eye(5); C(:,3:5) = 0; Xc = X*C

C0 = eye(rank(X)+1) - C*pinv(C); X0 = X*C0;





GLM: contrasts

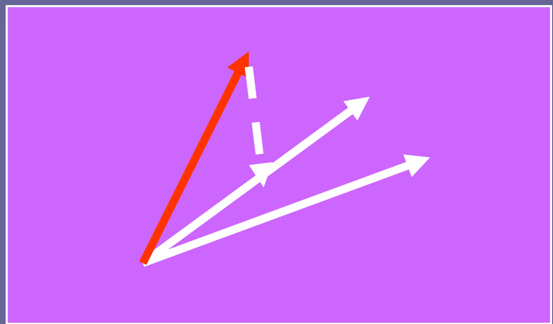
```
C = eye(5); C(:,3:5) = 0; Xc = X*C
C0 = eye(rank(X)+1) - C*pinv(C); X0 = X*C0;
var  = ((R*Y)'*(R*Y)) / dfe;
T = (C*Betas) ./ sqrt(var.*(C*pinv(X'*X)*C'));
```

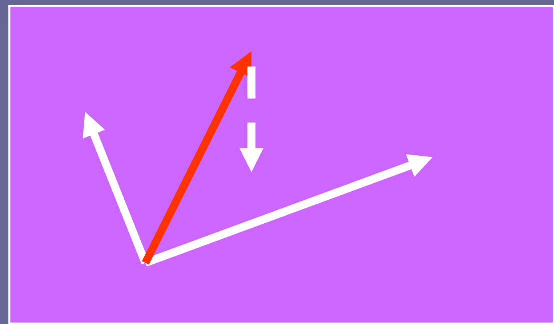
→ In SPM, you specify your design matrix and then you enter your contrasts

→ The T test is unilateral (A>B ≠ A<B)

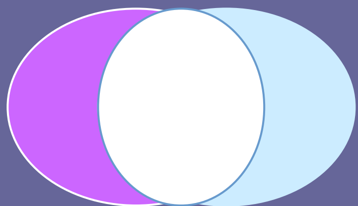
Orthogonalization

- It sometimes happens that regressors have to be orthogonalized (~PCA)

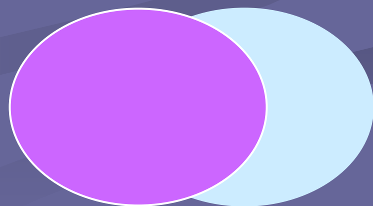




Lot of variance shared - because we look for the unique part of variance, the shared part goes into the error



Orthogonalization removes shared variance BUT order matters ! (like step by step regression)



Orthogonalization

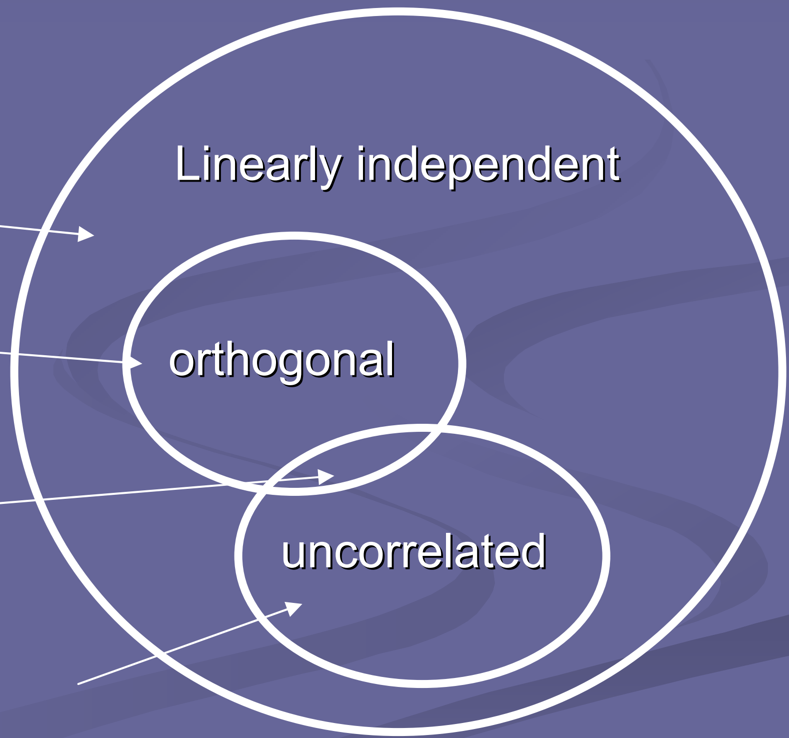
- Linearly independent ($Y \neq aX$), orthogonal ($X'Y = 0$) and uncorrelated ($(X\text{-mean}(X))'(Y\text{-mean}(Y))=0$) variables

[1 1 2 3] and [2 3 4 5]
Independent, correlated, not orthogonal

[1 -5 3 -1] and [5 1 1 3]
Independent, correlated and orthogonal

[-1 -1 1 1] and [1 -1 1 -1]
Independent, uncorrelated and orthogonal

[0 0 1 1] and [1 0 1 0]
Independent, uncorrelated, not orthogonal



Intermediate summary 2

- Any statistics can be performed using the GLM
- The design matrix describes your experiment but is also the model of your statistics (gp, repeated measures, covariates, interactions …)
- GLM links arithmetic and geometry (nice), meaning that we can combine vectors to create subspaces (contrasts) to test various effects
- Regressors (vectors) have to be independent but can still be correlated and not orthogonal (although the more uncorrelatd and close to orthogonal the better)

General convolution model (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

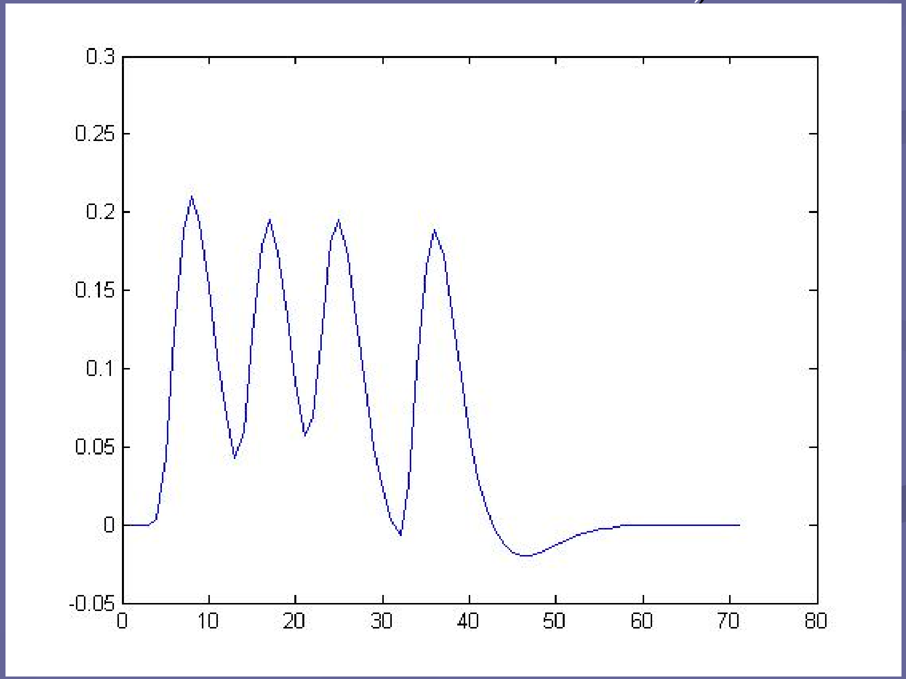
General linear convolution model

- **$X(y) = u(t) \otimes h(\tau)$**

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]; ← SPM onsets / user's job

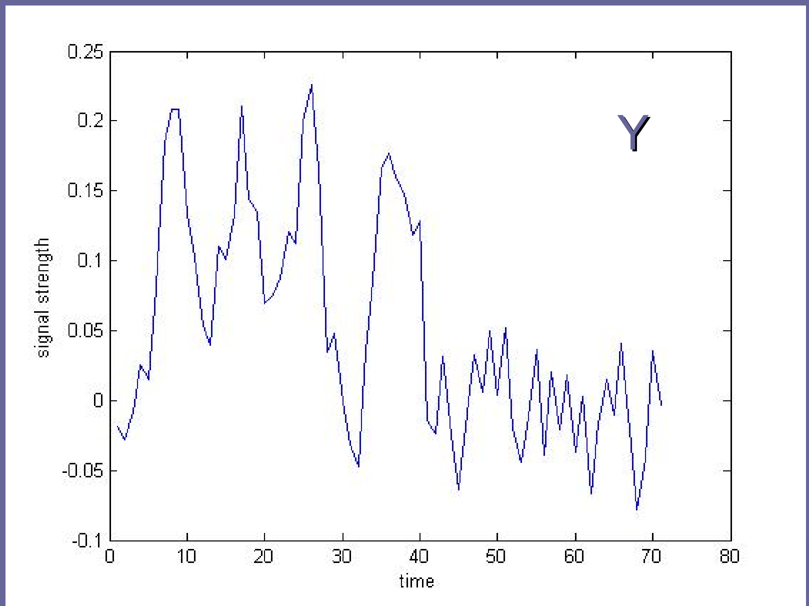
h = spm_hrf(1)

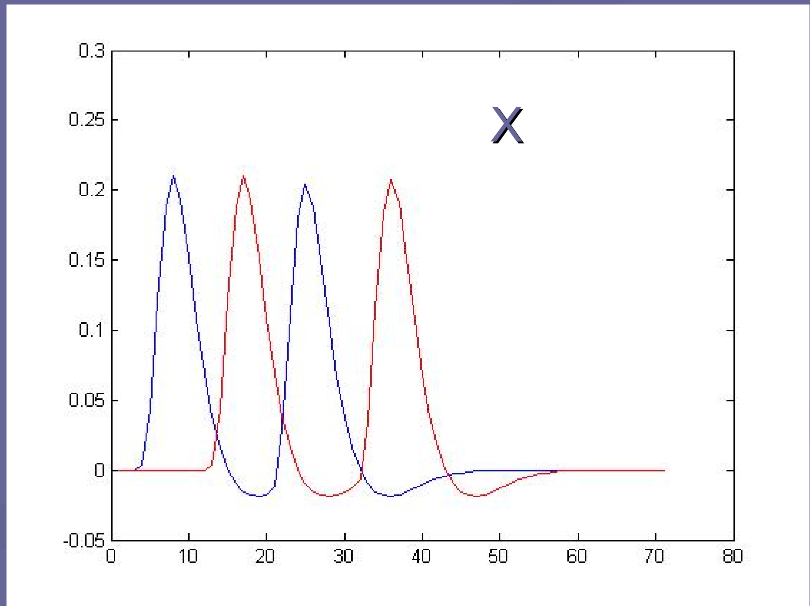
X = conv(u,h);



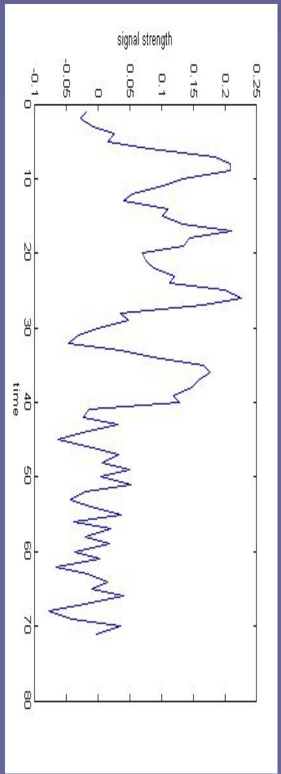
General linear convolution model

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

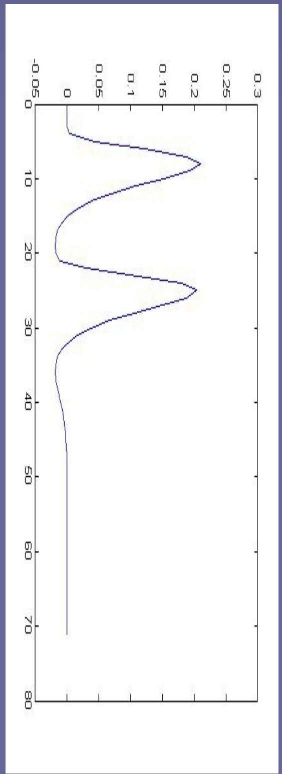




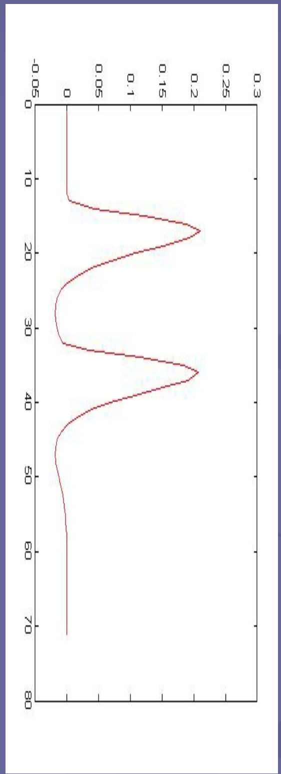
General linear convolution model



=



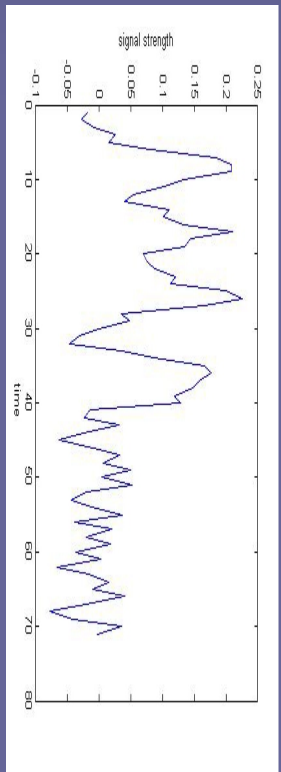
β1 +



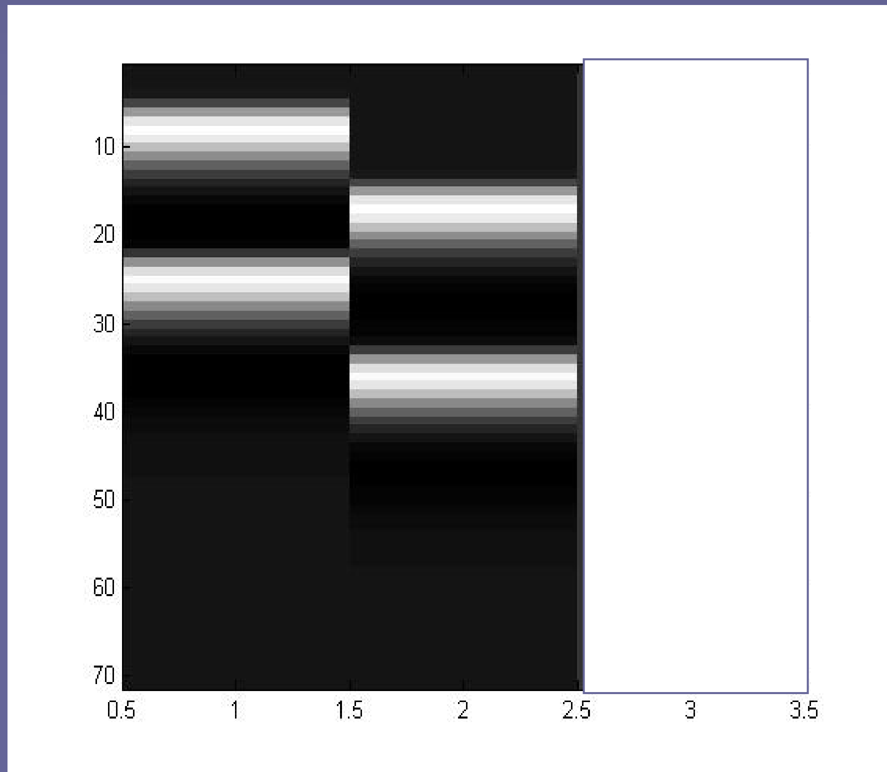
β2 + u + e

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

General linear convolution model



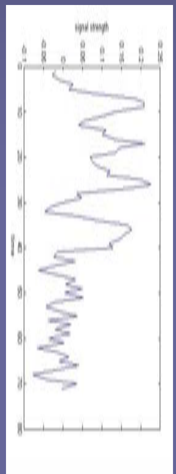
=



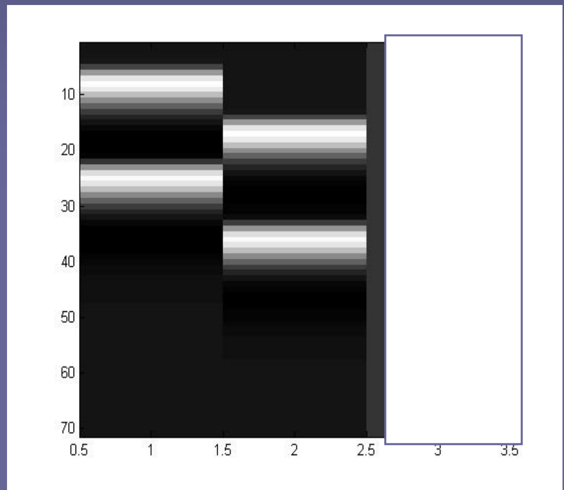
β1
β2 + e
u

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

General linear convolution model



=



$\beta 1$
$\beta 2$ + e
u

Your fMRI data

A set a beta.img / hrd created by SPM

SPM.mat → created by you by entering the different conditions

Final Summary

General linear model

- Y=XB+e is the master equation
- Describing X is the most important thing – you can model any design (Regression, ANOVA, ANCOVA, etc)
- ?? Well having good fMRI Y data matters too ☺
- SPM uses a convolution but after all, you can still thing about this as a linear regression