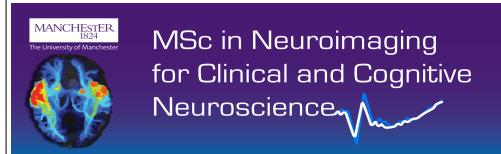




# Non-parametric Approaches to Inference

Dr. Martyn McFarquhar

Division of Neuroscience & Experimental Psychology  
The University of Manchester



Edinburgh SPM course, 2017



## The parametric approach

The **SPM** approach is based on traditional **parametric statistics**

We makes **assumptions** about the **form** and **parameters** of the **population distribution** of the data

$$y \sim \mathcal{N}(\mu, \sigma^2)$$

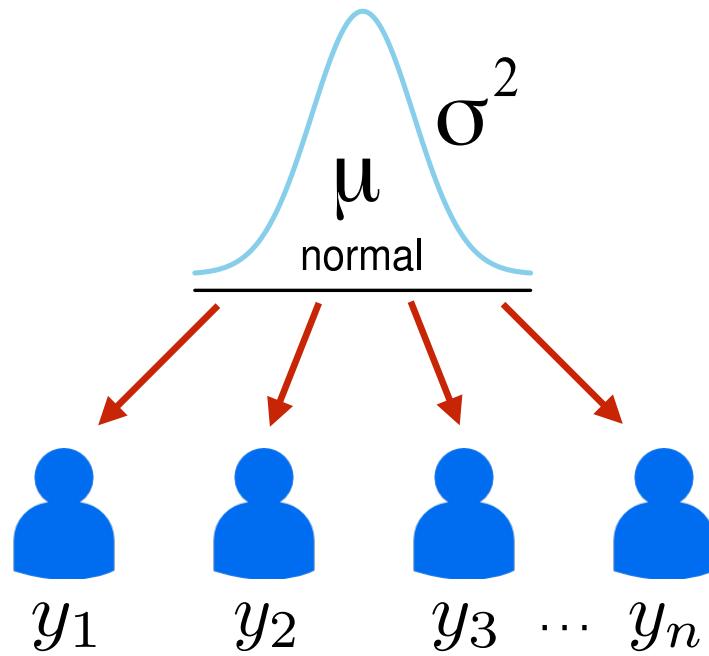
$$y \sim \text{Pois}(\lambda)$$

**Non-parametric** methods make no such assumptions

Important to understand the **parametric** approach in order to understand how **non-parametric** methods differ



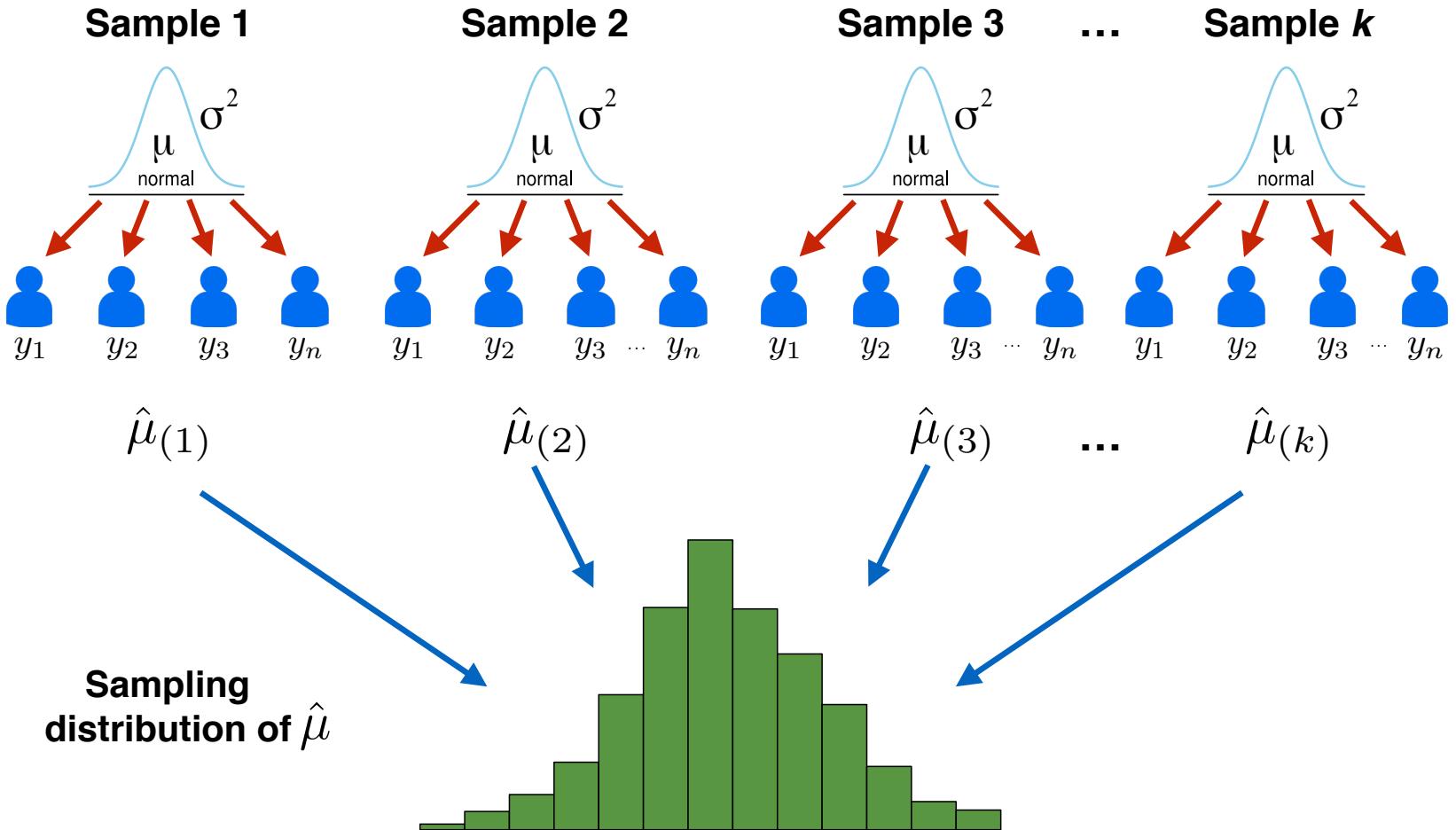
## The parametric approach



In the **classical** statistical framework the **parameters** of the population distribution are **constants** that we wish to **estimate**



# The parametric approach





## The parametric approach

The **sampling distribution** is the **probability distribution** of a **statistic** estimated from a **sample** of size  $n$

It can tell us the **most probable** values of  $\hat{\mu}$  and how **variable** those values are

If we **assume** that the **population distribution** is **normal** then the **sampling distribution** of  $\hat{\mu}$  is easily derived

$$\hat{\mu} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

As the sample mean is **unbiased** its **average value** is the **true population value**

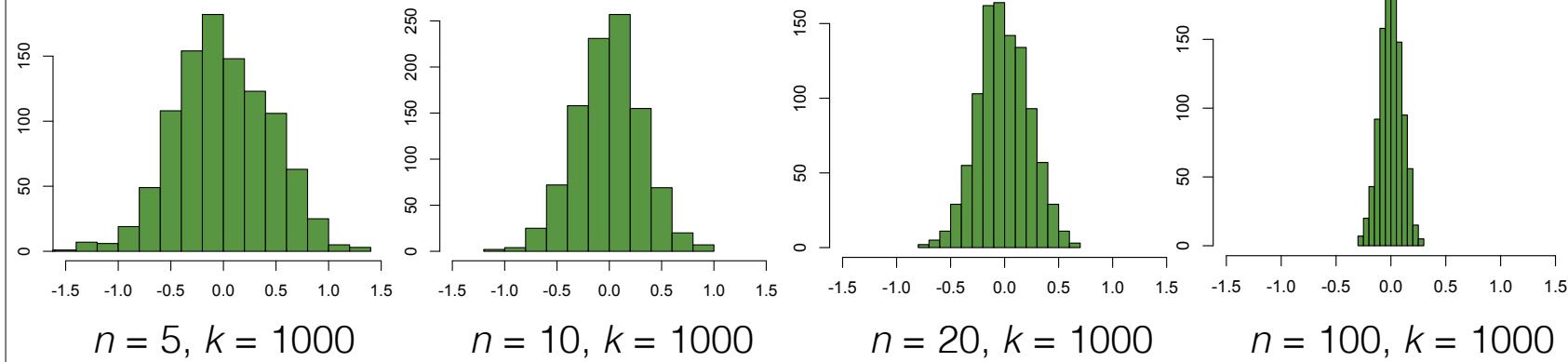


## The parametric approach

$$\hat{\mu} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

Although our interest in the **true value** of  $\mu$ , all we know from the **sampling distribution** is that

- On **average** the sample mean will equal the true value
- Larger samples lead to **lower variance/higher precision** of the estimate





## The parametric approach

In order to make a decision about the **true value** of  $\mu$  we perform a **hypothesis test**

This involves comparing our **estimate** of  $\mu$  to some **proposed value** for the true  $\mu$  — usually taken as **0**

The **difference** is not very meaningful on its own as it depends on **how variable the estimate is**

If we **divide** the **difference** by its **variability**, we can produce a **standardised index** of how “**meaningful**” the difference is

$$z = \frac{\hat{\mu} - \mu}{\sqrt{\text{Var}(\hat{\mu})}} = \frac{\hat{\mu}}{\sqrt{\sigma^2/n}} \quad t = \frac{\hat{\mu} - \mu}{\sqrt{\widehat{\text{Var}}(\hat{\mu})}} = \frac{\hat{\mu}}{\sqrt{\hat{\sigma}^2/n}}$$



## The parametric approach

We are still left with a problem — how **big** does  $z$  or  $t$  need to be for us to **make a decision** about the proposed value of  $\mu$ ?

If we know the **sampling distribution** of the **estimate** of  $\mu$  (and the **estimate** of  $\sigma^2$ ) we can derive the **distribution** of  $z$  and  $t$  when the **true value** of  $\mu$  equals the **proposed value**

$$z \sim \mathcal{N}(0, 1)$$

$$t \sim T(n - 1)$$

This is the **null distribution** of the **test statistic**

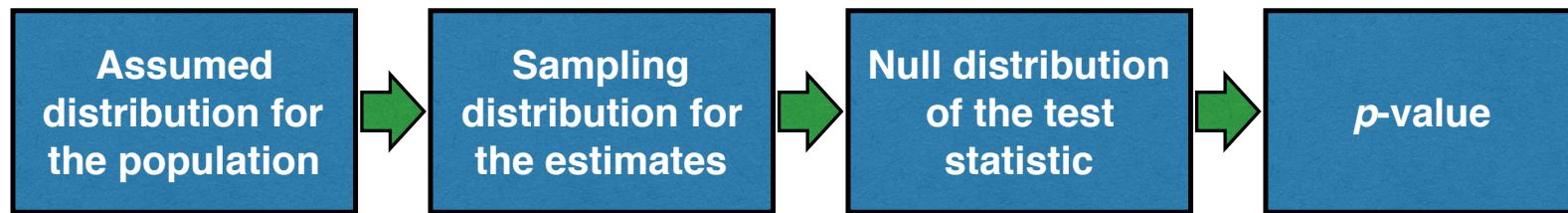
From this we can calculate the **p-value**

- **Probability** of finding a **test statistic** as large, or larger, **if the null were true**



# The parametric approach

The point is that

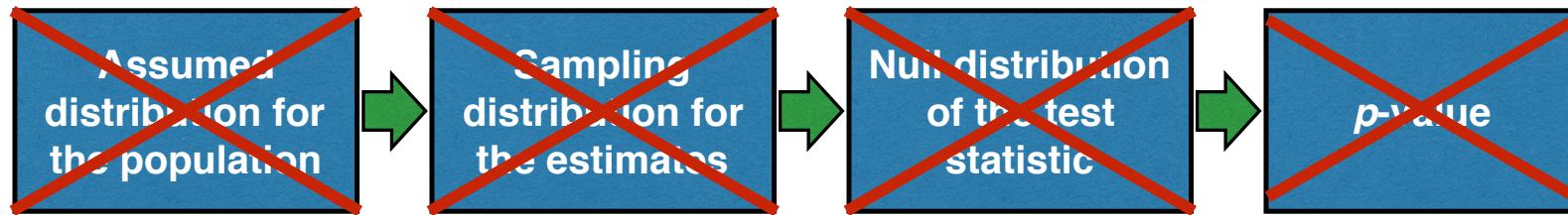


As such, if we use the **p-value** for **inference** we have to accept that its **validity** rests **entirely on the assumed distribution of the population**



# The parametric approach

The point is that



As such, if we use the **p-value** for **inference** we have to accept that its **validity** rests **entirely on the assumed distribution of the population**

If the **population distribution** is **far from normal** then

- The sampling distribution of the estimates will be **wrong**
- The null distribution of the test statistic will be **wrong**
- The **p-value** will be **wrong**



## Non-parametric approaches

The **validity** of **parametric** approaches rest on assumptions about the **distribution** in the **population**

**Non-parametric** approaches **do not** make any assumptions about the population distribution — their validity does not rely on assuming a probability model for the population

Range of non-parametric procedures that you may be aware of

- Friedman's ANOVA
- Kruskall-Wallis Test
- Mann-Whitney U
- Wilcoxon signed rank test

As computing power has **increased** the use of such tests has been somewhat overshadowed by **resampling methods**



# The resampling solution

There are a number of **different** resampling methods

- **Bootstrap**
- **Jackknife**
- **Permutation testing**

No matter the approach, the principle is the same

- We treat our **sample** as a **proxy** for the **population**
- Use a specific scheme to (**re**)**sample** values from our **sample**
- Do this **many times** recomputing values of interest to build up a **distribution** that can be used for **inference**

In **neuroimaging** the dominant approach is becoming **permutation tests** — the most straightforward way of arriving at non-parametric versions of the  $p$ -value



## The permutation test

**Permutation testing** was originally developed by Fisher & Pitman during the 1930s

Use our data to **approximate the null** and thus calculate  $p$ -values **without assuming** any form for the **population distribution**

The use of **parametric** methods came about because it was **impractical** to perform permutation tests without computers





## The permutation test

**Permutation testing** was originally developed by Fisher & Pitman during the 1930s

Use our data to **approximate the null** and thus calculate *p*-values **without assuming** any form for the **population distribution**

The use of **parametric** methods came about because it was **impractical** to perform permutation tests without computers

“If Fisher...had access to modern computers, it is likely that permutation tests would be the standard procedure”

- Salkind (2010) *Encyclopedia of Research Design*





# The permutation test

## Example of two groups

The **key concept** in permutation testing is **rearranging** the **data labels**

Original data

3	A
6	A
9	A
7	B
4	B
1	B

**Mean of A = 6**

**Mean of B = 4**

**Difference = 2**

Permuted data

3	B
6	A
9	B
7	A
4	B
1	A

**Mean of A = 4.66**

**Mean of B = 5.33**

**Difference = 0.67**



# The permutation test

## Example of two groups

For **two groups** of **three subjects** there are **20** different orderings of the group labels

$$\binom{6}{3} = \frac{6!}{3!(6-3)!} = 20$$

For **each rearrangement** we calculate the **statistic of interest** and **save** it

- |            |            |
|------------|------------|
| 1. AAABBB  | 11. BAAABB |
| 2. AABABB  | 12. BAABAB |
| 3. AABBAB  | 13. BAABBA |
| 4. AABBBA  | 14. BABAAB |
| 5. ABAABB  | 15. BABABA |
| 6. ABABAB  | 16. BABBAA |
| 7. ABABBA  | 17. BBAAAB |
| 8. ABBAAB  | 18. BBAABA |
| 9. ABBABA  | 19. BBABAA |
| 10. ABBBAA | 20. BBBAAA |



# The permutation test

## Example of two groups

For **two groups** of **three subjects** there are **20** different orderings of the group labels

$$\binom{6}{3} = \frac{6!}{3!(6-3)!} = 20$$

For **each rearrangement** we calculate the **statistic of interest** and **save** it

**Importantly** the **original order** is included

- |            |            |
|------------|------------|
| 1. AAABBB  | 11. BAAABB |
| 2. AABABB  | 12. BAABAB |
| 3. AABBAB  | 13. BAABBA |
| 4. AABBBA  | 14. BABAAB |
| 5. ABAABB  | 15. BABABA |
| 6. ABABAB  | 16. BABBAA |
| 7. ABABBA  | 17. BBAAAB |
| 8. ABBAAB  | 18. BBAABA |
| 9. ABBABA  | 19. BBABAA |
| 10. ABBBAA | 20. BBBAAA |



# The permutation test

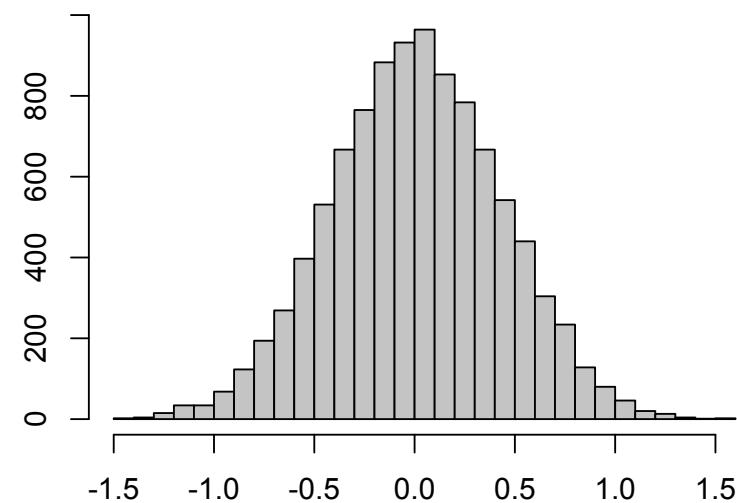
## Example of two groups

By **rearranging** the labels we are calculating the **test statistic** under the **null hypothesis of no difference**

By **saving** each value we build an **approximate null distribution**

Once we have this distribution we can calculate the **probability of getting our original test statistic, or larger, under the null**

This gives us an **approximate *p*-value**





# The permutation test

## Rearranging the data

Generally speaking it is not the **labels** that we rearrange, rather it is the **data itself**

**Original data**

3	A
6	A
9	A
7	B
4	B
1	B

$$\begin{aligned}\text{Mean of A} &= 6 \\ \text{Mean of B} &= 4\end{aligned}$$

**Permuted data**

6	A
7	A
1	A
3	B
9	B
4	B

$$\begin{aligned}\text{Mean of A} &= 4.66 \\ \text{Mean of B} &= 5.33\end{aligned}$$

For simple models of **groups**  
this is the same as rearranging  
the labels



# The permutation test

## Rearranging the data

Generally speaking it is not the **labels** that we rearrange, rather it is the **data itself**

**Original data**

18	<b>10</b>
26	<b>10</b>
34	<b>10</b>
17	<b>11</b>
28	<b>11</b>
14	<b>12</b>

**Slope** = -5.5

**Permuted data**

17	<b>10</b>
34	<b>10</b>
26	<b>10</b>
18	<b>11</b>
14	<b>11</b>
28	<b>12</b>

**Slope** = -1

For simple models of **groups**  
this is the same as rearranging  
the labels

**More general** — allows us to  
permute models with **no labels**



# The permutation test

## Conceptual justification for permutations

The **null hypothesis** under rearrangement is that there is **no relationship** between the **outcome variable** and the **predictor variables**

As such, the **pairing** of outcome values and predictor values is **arbitrary**

By **breaking** that pairing through **rearrangement** we create a **realisation** of the claim under the null

As such, we can see what values we would get **if those pairings were arbitrary**

Our ability to do this **relies** on assumptions of **exchangeability**



# The permutation test

## Exchangeability

Formally, **exchangeability** means that **under the null the joint distribution** of the data **do not change** under rearrangement

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \end{aligned}$$

**Any ordering** of  $\mathbf{Y}$  will lead to errors that are distributed as above

Designs that this precludes:

- Groups with **difference variances**
- **Repeated measures/non-independent** data

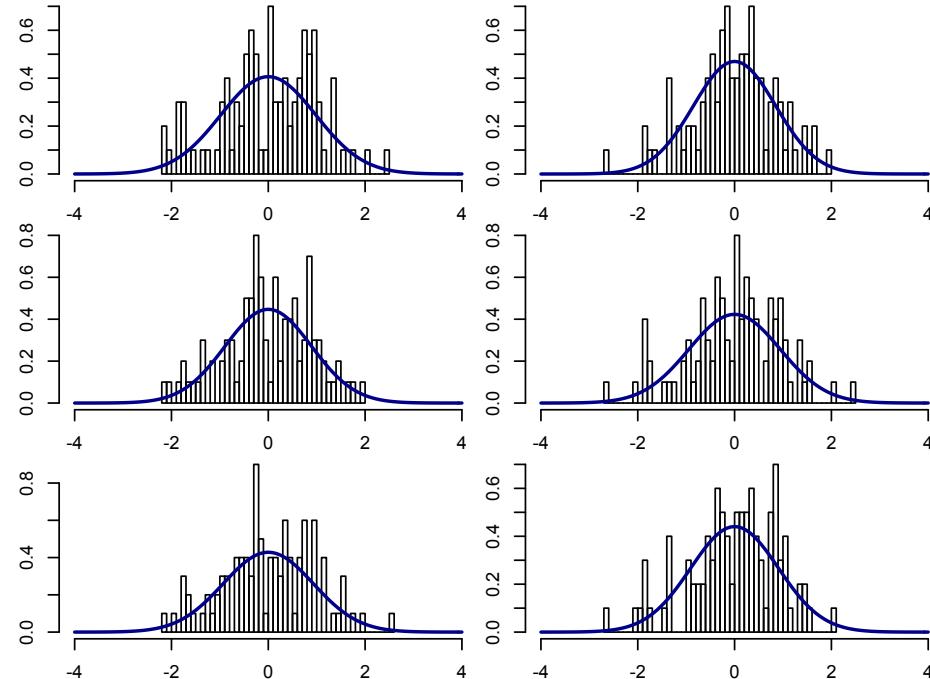
These can be incorporated using assumptions of **weak exchangeability**



# The permutation test

## Exchangeability — equal variance

Original error distributions  
for group **A** and **B**



Error distribution for **row  
swapping 1**

Error distribution for **row  
swapping 2**

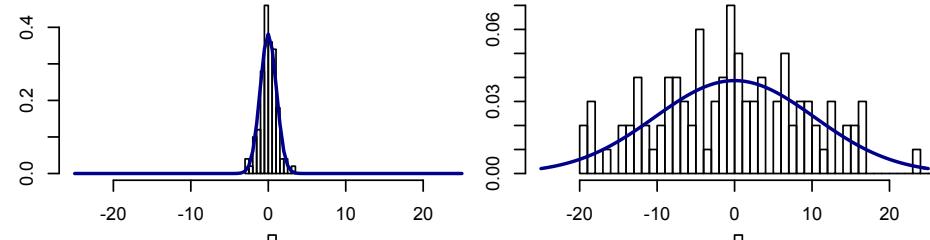
Under rearrangement the errors are conceivably drawn from the  
**same distribution** as the original groups



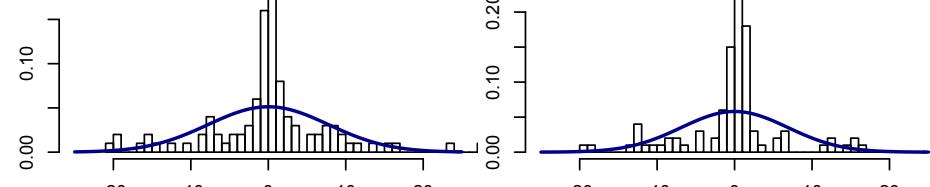
# The permutation test

## Exchangeability — unequal variance

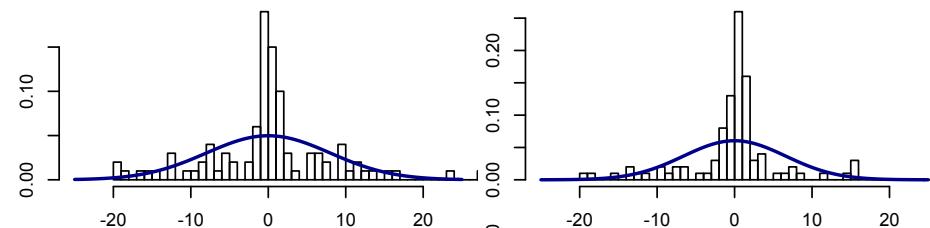
Original error distributions  
for group **A** and **B**



Error distribution for **row swapping 1**



Error distribution for **row swapping 2**



Under rearrangement the errors appear to be drawn from a **different distribution** to the original groups



# The permutation test

## Exchangeability

The **traditional** requirement for permutations is **exchangeable errors (EE)**

Under **rearrangement** of the data, the **distribution** of the **errors** remains the **same**

For any  $\mathbf{P}_j \in \mathcal{P}$ ,  $\boldsymbol{\epsilon} \stackrel{d}{=} \mathbf{P}_j \boldsymbol{\epsilon}$

Relaxes usual *i.i.d.* normal assumptions

- No need for a specific distribution
  - Must be **identical** across the errors
- More general than **independence** (see Winkler *et al.*, 2014)



# The permutation test

## Exchangeability

A **less restrictive** assumption is **independent and symmetric errors (ISE)** — allows for errors that are **not exchangeable**

Under **sign-flipping** of the data, the **distribution** of the **errors** remains the **same**

For any  $\mathbf{S}_j \in \mathcal{S}$ ,  $\epsilon \stackrel{d}{=} \mathbf{S}_j \epsilon$

Relaxes usual *i.i.d.* normal assumptions

- No need for a specific distribution
  - Must be **identical and symmetric**
- **Independence** is a necessity

If **both** EE and ISE we can **row-swap and sign-flip**

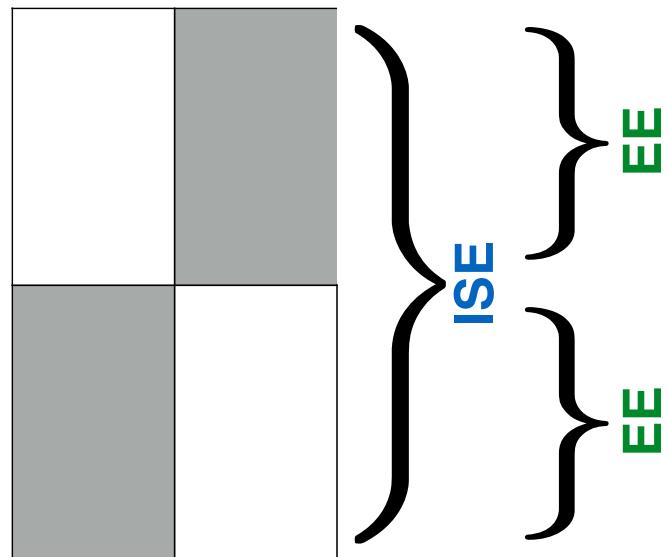


# The permutation test

## Weak exchangeability

Sometimes neither **EE** nor **ISE** can be assumed for the data as a **whole** — with a **blocked structure** then **EE** or **ISE** can be assumed **at the level of the blocks**

*Example: Difference variances*



### Within block exchangeability

Note that **row-swapping within each group** will have no effect on the **group means**

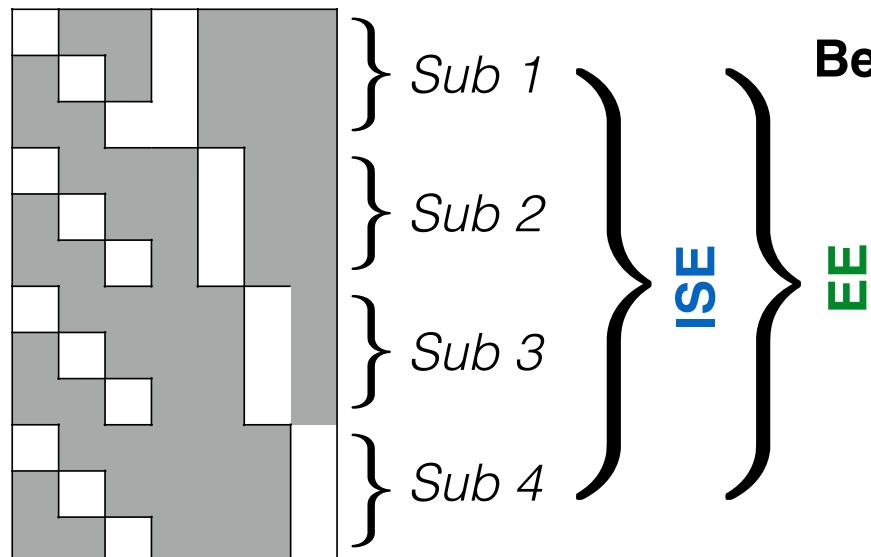


# The permutation test

## Weak exchangeability

Sometimes neither **EE** nor **ISE** can be assumed for the data as a **whole** — with a **blocked structure** then **EE** or **ISE** can be assumed **at the level of the blocks**

*Example: Repeated measures*



### Between block exchangeability

**Cannot** swap rows or perform sign flips **within a subject**

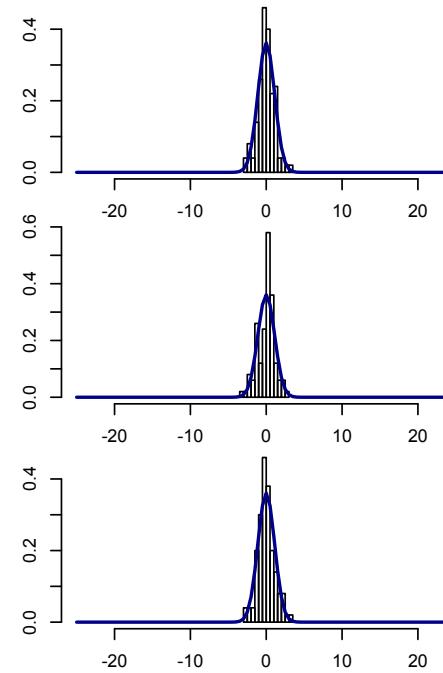
**Can** swap rows and sign flip **subjects as a whole**



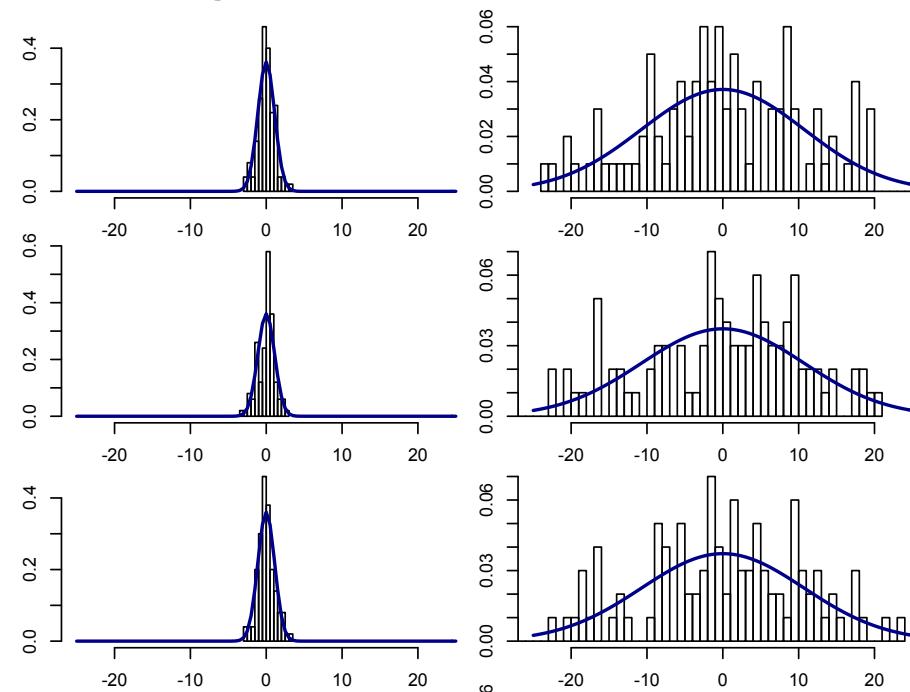
# The permutation test

## Sign flipping for different variance groups

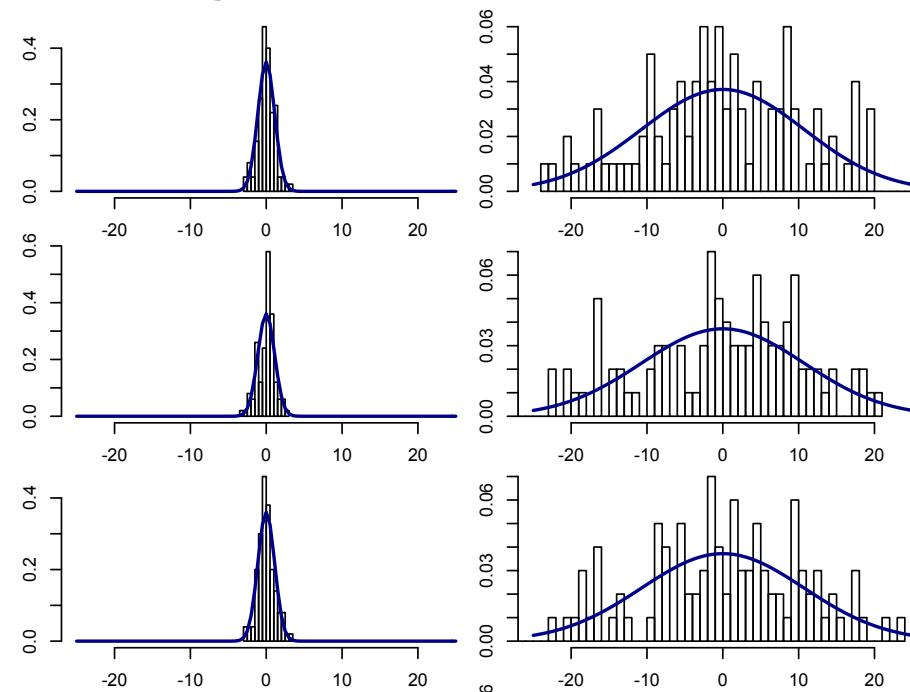
Original error distributions  
for group **A** and **B**



Error distribution for **sign-flipping 1**



Error distribution for **sign-flipping 2**



By using **sign-flipping alone** we can now conceive of the errors as drawn from the **same distributions** as the original data



# The permutation test

## Sign flipping for different variance groups

### R Code

```
Y1 <- rnorm(100, 10, 1)
Y2 <- rnorm(100, 10, 5)

Y      <- c(Y1,Y2)
Groups <- as.factor(c(rep("A",100), rep("B",100)))

paraTval <- t.test(Y ~ Groups)$stat

tdist <- rep(0,10000)

for (i in 1:10000){
  signFlips <- sample(c(-1,1), size=200, replace=TRUE)
  newY      <- signFlips*Y
  tdist[i]   <- t.test(newY ~ Groups)$stat
}

permPval <- mean(abs(tdist) >= paraTval)
```

```
> permPval
[1] 0.3407

> t.test(Y ~ Groups)

Welch Two Sample t-test

data: Y by Groups
t = 0.94409, df = 107.24, p-value = 0.3472
```

Random sign-flipping is enough to **break the connection** between the outcome and predictors **without rearrangement**



# Permutation testing in the GLM

## Permutation matrix

We can **rearrange** and/or **sign-flip** the data by **pre-multiplying** with a permutation matrix

$$\begin{array}{c}
 \text{Permutation} \\
 \text{matrix} \\
 \hline
 \begin{matrix}
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
 \times
 \begin{array}{c}
 \text{Original} \\
 \text{data} \\
 \hline
 \begin{matrix}
 -4.01 \\
 -2.97 \\
 -1.98 \\
 -1.01 \\
 0.00 \\
 +0.96 \\
 +2.03 \\
 +3.00 \\
 +3.00 \\
 +4.02
\end{matrix}
\end{array}
 =
 \begin{array}{c}
 \text{Permuted} \\
 \text{data} \\
 \hline
 \begin{matrix}
 -1.98 \\
 +2.03 \\
 0.00 \\
 +4.02 \\
 -2.97 \\
 +0.96 \\
 +3.00 \\
 -4.01 \\
 -1.01
\end{matrix}
\end{array}
 \quad
 \begin{array}{c}
 \text{Sign-flipping} \\
 \text{matrix} \\
 \hline
 \begin{matrix}
 -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{matrix}
\end{array}
 \times
 \begin{array}{c}
 \text{Original} \\
 \text{data} \\
 \hline
 \begin{matrix}
 -4.01 \\
 -2.97 \\
 -1.98 \\
 -1.01 \\
 0.00 \\
 +0.96 \\
 +2.03 \\
 +3.00 \\
 +3.00 \\
 +4.02
\end{matrix}
\end{array}
 =
 \begin{array}{c}
 \text{Sign-flipped} \\
 \text{data} \\
 \hline
 \begin{matrix}
 +4.01 \\
 +2.97 \\
 -1.98 \\
 -1.01 \\
 0.00 \\
 +0.96 \\
 -2.03 \\
 +3.00 \\
 +3.00 \\
 -4.02
\end{matrix}
\end{array}$$

The **original** data can be seen as **pre-multiplication** with an **identity permutation matrix**

Winkler *et al.* (2014) *NeuroImage*



# Permutation testing in the GLM

## Permutation matrix

We can **rearrange** and/or **sign-flip** the data by **pre-multiplying** with a permutation matrix

Permutation and sign-flipping matrix									Original data	Permuted & sign-flipped
0	0	+1	0	0	0	0	0	0		
0	0	0	0	0	0	-1	0	0	-4.01	-1.98
0	0	0	0	-1	0	0	0	0	-2.97	-2.03
0	0	0	0	0	0	0	0	-1.98	0.00	
0	0	0	0	0	0	0	0	-1.01	-4.02	
0	-1	0	0	0	0	0	0	0.00	+2.97	
0	0	0	0	0	+1	0	0	+0.96	+0.96	
0	0	0	0	0	0	0	+1	+2.03	+3.00	
-1	0	0	0	0	0	0	0	+3.00	+4.01	
0	0	0	+1	0	0	0	0	+4.02	-1.01	

Winkler *et al.* (2014) *NeuroImage*



# Permutation testing in the GLM

## Permutation matrix

0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1

Block 1      Block 2      Block 3

0	0	0	1	0	0	0	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	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1
1	0	0	0	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

-1	0	0	0	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	0	+1	0	0	0	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	0	-1	0	0	0
0	0	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	-1	0

**Within-block  
exchangeability**

**Whole-block  
exchangeability**

Winkler *et al.* (2014) *NeuroImage*



# Permutation testing in the GLM

## Permutation matrix

The simplest form of a **permuted** GLM would then be

$$\mathbf{P}\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

Assuming that there are **no nuisance covariates in X**

Permutation strategy

1. Calculate the **reference test statistic** using  $\mathbf{P} = \mathbf{I}$
2. Draw a random  $\mathbf{P}$  and calculate the **test statistic** again
3. If the **new statistic** is **greater than or equal** to the **reference** we **count it**
4. Repeat **2** and **3** many times
5. Divide the **number of counts + 1** by the **number of permutations + 1** to get our **p-value**



# Permutation testing in the GLM

## Example in MATLAB

```
% Generate some data
Y(1:20) = normrnd(6,2,20,1);
Y(21:40) = normrnd(5,2,20,1);

% Design matrix and contrast
X = [ones(1,20) zeros(1,20); zeros(1,20) ones(1,20)]';
L = [1 -1];

% Reference model
beta = inv(X'*X)*X'*Y;
resid = Y - X*beta;
sigma = (resid'*resid) / 38;
refF = ((L*beta)' * inv(L*inv(X'*X)*L') * (L*beta)) / sigma;

% Parametric p-value
paraP = fcdf(refF, 1, 38, 'upper');
```

Randomly generate some data, fit a two-sample GLM with a contrast comparing the groups using an  $F$ -test



# Permutation testing in the GLM

## Example in MATLAB

```
% Permutations
count = zeros(5000,1);
for i = 1:5000
    P          = eye(40);
    P          = P(randperm(40),:);
    Yperm      = P*Y;
    betaPerm   = inv(X'*X)*X'*Yperm;
    residPerm  = Yperm - X*betaPerm;
    sigmaPerm  = (residPerm'*residPerm) / 38;
    Fdist(i,1) = ((L*betaPerm)' * inv(L*inv(X'*X)*L') * (L*betaPerm)) / sigmaPerm;
    if Fdist(i,1) >= refF
        count(i) = 1;
    end
end

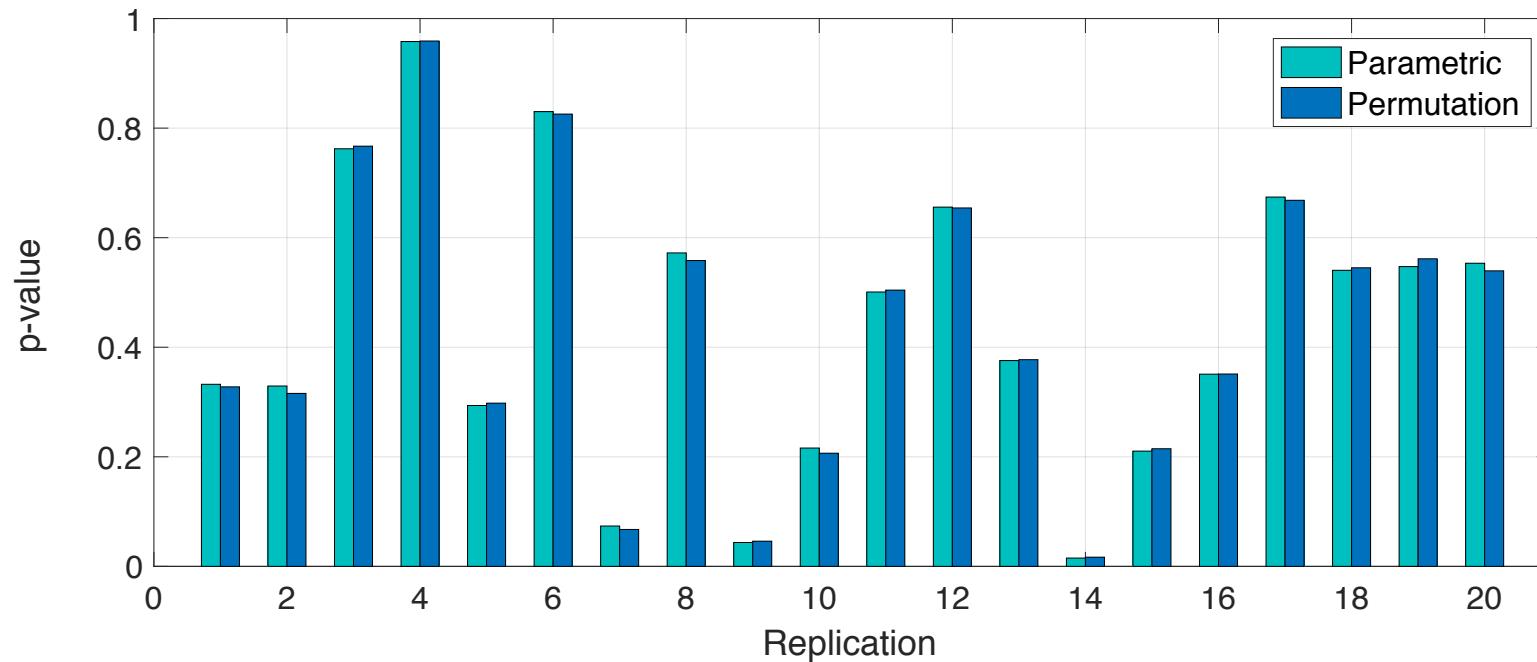
% Permutation p-value
permP = (sum(count) + 1) / (5001);
```

Across 5,000 repetitions create a random permutation matrix,  
pre-multiply the data, re-fit the model and recalculate the  $F$ -test



# Permutation testing in the GLM

## Example in MATLAB



**Strong correspondence** between **parametric** and **non-parametric**  $p$ -values when data are **normal**



# Permutation testing in the GLM

## Permutation testing with nuisance covariates

So far we have only considered models with a **single continuous covariate** or **single factor**

In cases where we have **multiple** covariate/factors permutations become more difficult

Under the null of **no effect** for one covariate/factor there may still be **other effects** in the model that **should be accounted for**

If we just permuted the data, we will disrupt **all the effects** in the model — not looking at the model under the null of one effect being 0 with **all other effects accounted for**



# Permutation testing in the GLM

## Permutation testing with nuisance covariates

The Freedman-Lane (Freedman & Lane, 1983) approach is

$$\mathbf{Y}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

**Effects of interest**      **Nuisance effects**

The diagram shows the equation  $\mathbf{Y}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$ . Two terms,  $\mathbf{X}\boldsymbol{\beta}$  and  $\mathbf{Z}\boldsymbol{\gamma}$ , are circled:  $\mathbf{X}\boldsymbol{\beta}$  is circled in green and labeled "Effects of interest" with a green arrow pointing to it;  $\mathbf{Z}\boldsymbol{\gamma}$  is circled in red and labeled "Nuisance effects" with a red arrow pointing to it.

The permuted data  $\mathbf{Y}^*$  are given by

$$\mathbf{Y}^* = \mathbf{P}\boldsymbol{\epsilon}_{\mathbf{Z}} + \mathbf{Z}\boldsymbol{\gamma}$$

Residuals from the model containing  
only the nuisance effects

Can also express the model as

$$(\mathbf{P}\mathbf{R}_{\mathbf{Z}} + \mathbf{H}_{\mathbf{Z}})\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$



# Permutation testing in the GLM

## Randomise algorithm

The key steps in the `randomise` algorithm are:

1. The **contrast** is used to **partition** the model into **effects of interest** and **nuisance effects**
2. The model containing **only nuisance effects** is fit and **nuisance partition residuals** are formed
3. These residuals are **permuted**
4. The **nuisance effects** are added back, the **full model** is fit and the **test statistic** is calculated and saved
5. Steps 3 and 4 are repeated many times to build up the **null distribution** of the test statistic



# Permutation tests in neuroimaging

## Multiple comparisons correction

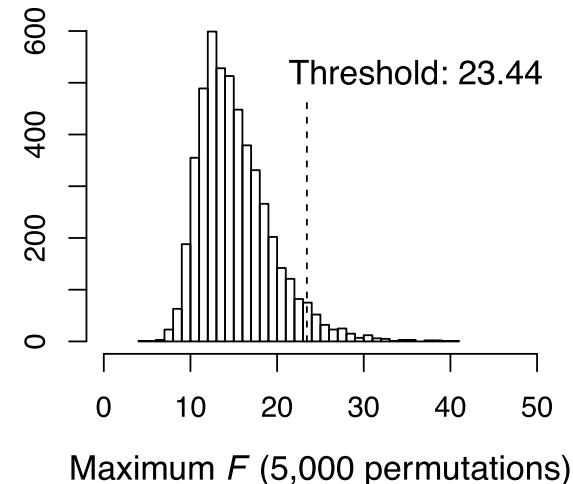
Permutation methods can also provide a **family wise error correction** as an alternative to RFT

For each iteration we

1. Calculate the **largest** statistic in the image
2. For each voxel compare the **reference** statistic to the largest permuted statistic

Build the null distribution of **largest test statistics**

The  $p$ -value is then the probability of finding a statistic as large, or larger, under the null **anywhere in the image**





# Permutation tests in neuroimaging

## Multiple comparisons correction

Though this is naturally suited to **voxel inference** we can extend the approach to **cluster inference** as well

For each iteration we

1. Form **clusters** by thresholding the image using **parametric p-values**
2. Find the **largest** cluster in the image
3. Compare the size of each **reference cluster** to the largest permuted cluster

Build the null distribution of **largest clusters**

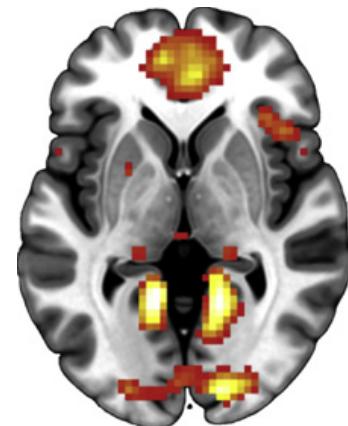
The  $p$ -value is then the probability of finding a **cluster** as large, or larger, under the null **anywhere in the image**

# Permutation tests in neuroimaging

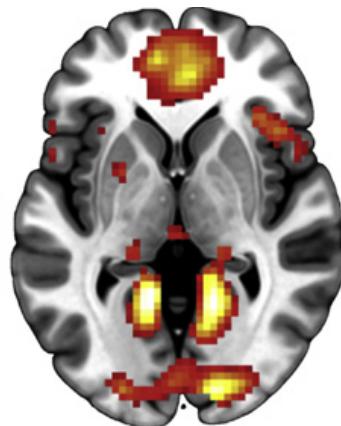
## Comparison between permutations and GRF

Adapted version of `randomise` given in McFarquhar *et al.* (2016)

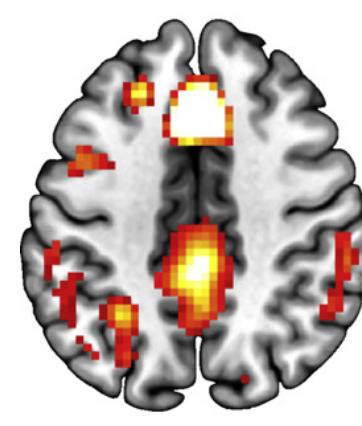
5691



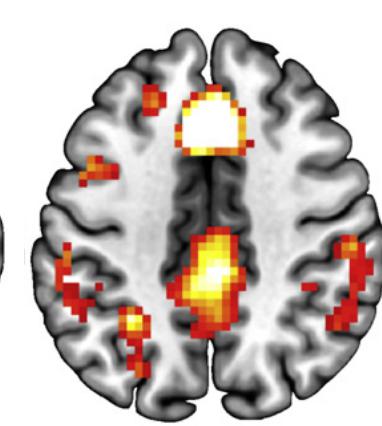
7151



5733



5906



**GRF**

**Perm.**

**GRF**

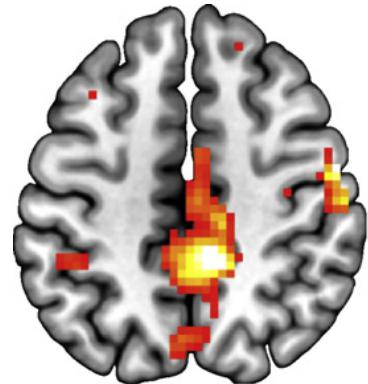
**Perm.**

# Permutation tests in neuroimaging

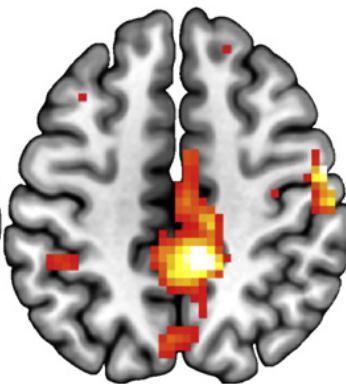
## Comparison between permutations and GRF

Adapted version of `randomise` given in McFarquhar *et al.* (2016)

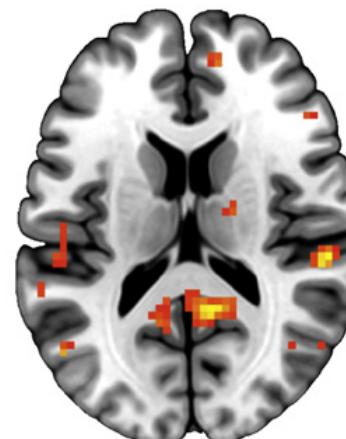
2725

**GRF**

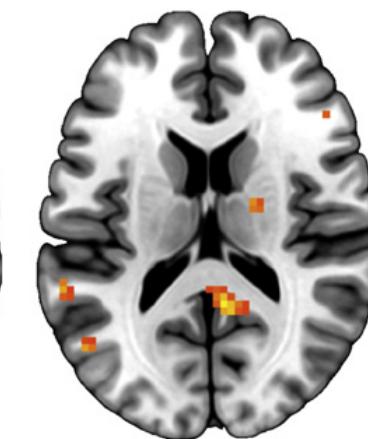
2725

**Perm.**

1601

**GRF**

5906

**Perm.**



# Permutation tests in neuroimaging

**Eklund, Nichols & Knutsson (2016)**

The advantages of **permutation methods** was recently put into **sharp focus** by Eklund *et al.* (2016)

Took **resting state** dataset as **null data** and fit **blocked** and **event-related** designs using **SPM**, **FSL** and **AFNI**

Compared **cluster** and **voxel** correction techniques across 1,000 repetitions of the analysis using

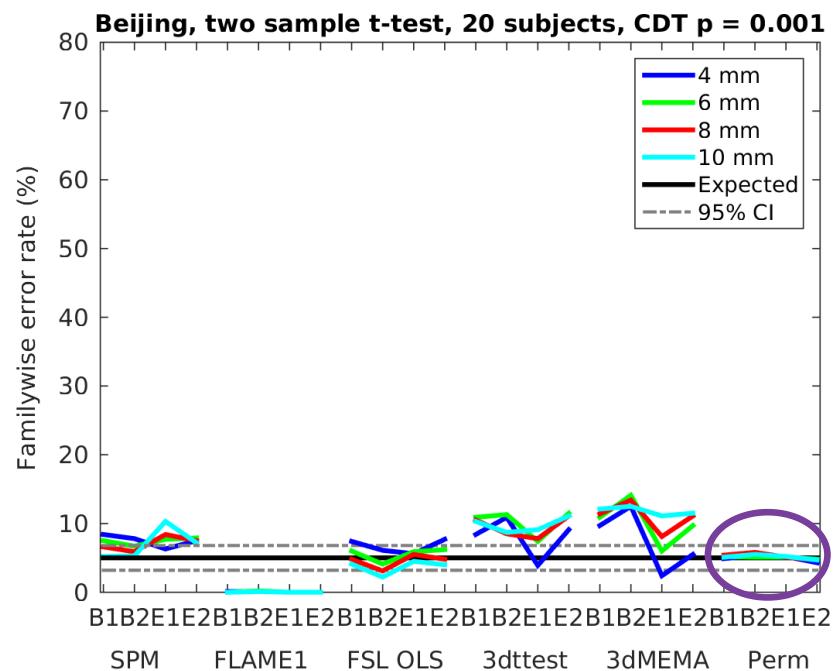
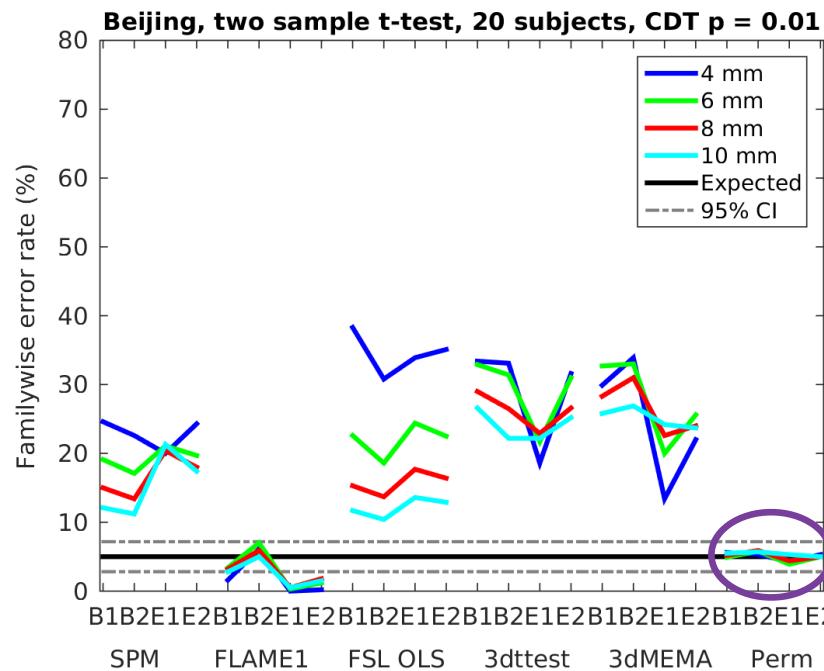
- GRF (SPM and FSL)
- Monte Carlo simulation (AFNI)
- Permutation tests

As the **model** and the **data** have **no correspondence** we should only see a **significant result anywhere in the brain** 5% of the time



# Permutation tests in neuroimaging

**Eklund, Nichols & Knutsson (2016)**

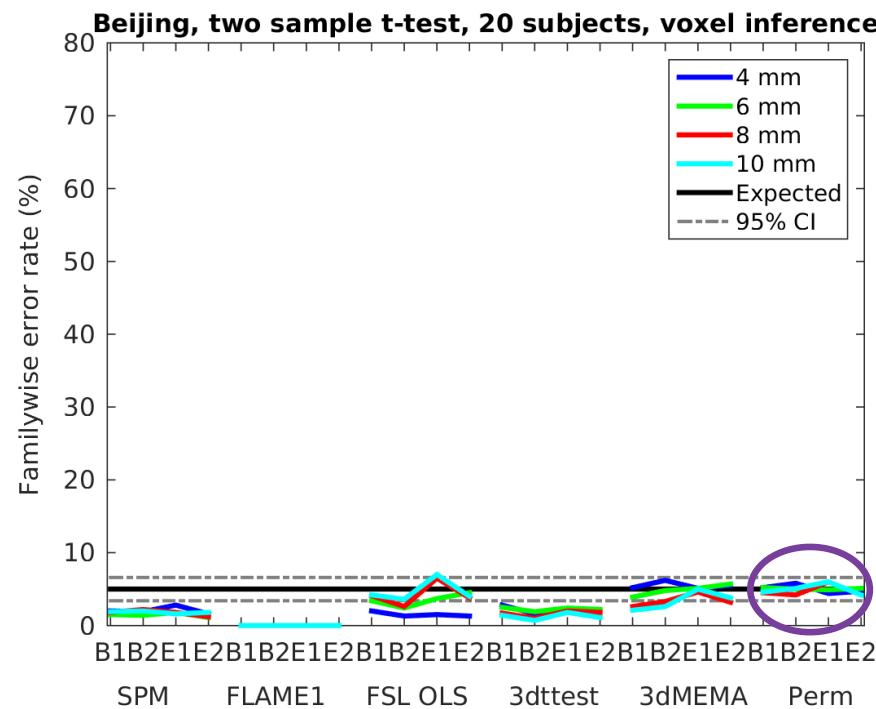


**Cluster-level correction**



# Permutation tests in neuroimaging

**Eklund, Nichols & Knutsson (2016)**



**Voxel-level correction**



# Permutation testing software

Permutation tests are **not available** in **SPM**

Need specific **toolboxes/software**

- **SnPM**
  - Tom Nichols & Andrew Holmes (Nichols & Holmes, 2001)
  - SPM Toolbox — ([warwick.ac.uk/snpm](http://warwick.ac.uk/snpm))
- **MRM**
  - Martyn McFarquhar (McFarquhar *et al.*, 2016)
  - MATLAB Toolbox ([www.click2go.umpip.com](http://www.click2go.umpip.com))
- **PALM**
  - Andersson Winkler
  - MATLAB Toolbox — ([fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM](http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM))
- **randomise**
  - Steve Smith, Tim Behrens, Matthew Webster & Tom Nichols (Winkler *et al.*, 2014)
  - FSL command line tool — ([fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL](http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL))



# Permutation tests in neuroimaging

## Computational burden

Consider what we are asking the computer to do

If we want **5,000** permutations, even refitting the model in **1 second** would mean waiting **1 hour 23 minutes** for it to finish

We need to get the computer to **re-estimate** the model **at every voxel** in much less than **1 second** to get decent time

GPGPU — my own experiments have yielded a model refit of ~23ms — about 2 minutes for 5,000 permutations

Others methods revolve around **simulating** the tails of the null (Winkler *et al.*, 2016)





## The future of neuroimaging

There seems little reason to not use **permutation tests** with neuroimaging datasets

We do not have to assume that normality holds **at every voxel**, and get a **multiple comparisons** correction that **always works**

We can extend the calculation of *p*-values to **non-standard statistics** irrespective of their tractability under the null

The challenge now is not **statistical** but **computational** — parallel processing seems the obvious approach

As computer power increases, permutation testing **will get faster** — at which point there are **no reasons** not to use it



## References

- Eklund, A., Nichols, T.E. & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false positive rates, *Proceedings of the National Academy of Sciences*, 113, 7900-05.
- McFarquhar, M., McKie, S., Emsley, R., Suckling, J., Elliott, R. & Williams, S. (2016). Multivariate and repeated measures (MRM): A new toolbox for dependent and multimodal group-level neuroimaging data, *NeuroImage*, 132, 373-89.
- Nichols, T.E. & Holmes, A.P. (2001). Nonparametric permutation tests for functional neuroimaging: A primer with examples, *Human Brain Mapping*, 15, 1-25.
- Winkler, A.M., Ridgway, G.R., Webster, M.A., Smith, S.M. & Nichols, T.E. (2014). Permutation inference for the general linear model, *NeuroImage*, 92, 381-97.
- Winkler, A.M., Ridgway, G.R., Douaud, G., Nichols, T.E. & Smith, S.M. (2016). Faster permutation inference in brain imaging, *NeuroImage*, 141, 502-16.