

Large Scale Inference

Lieven Clement

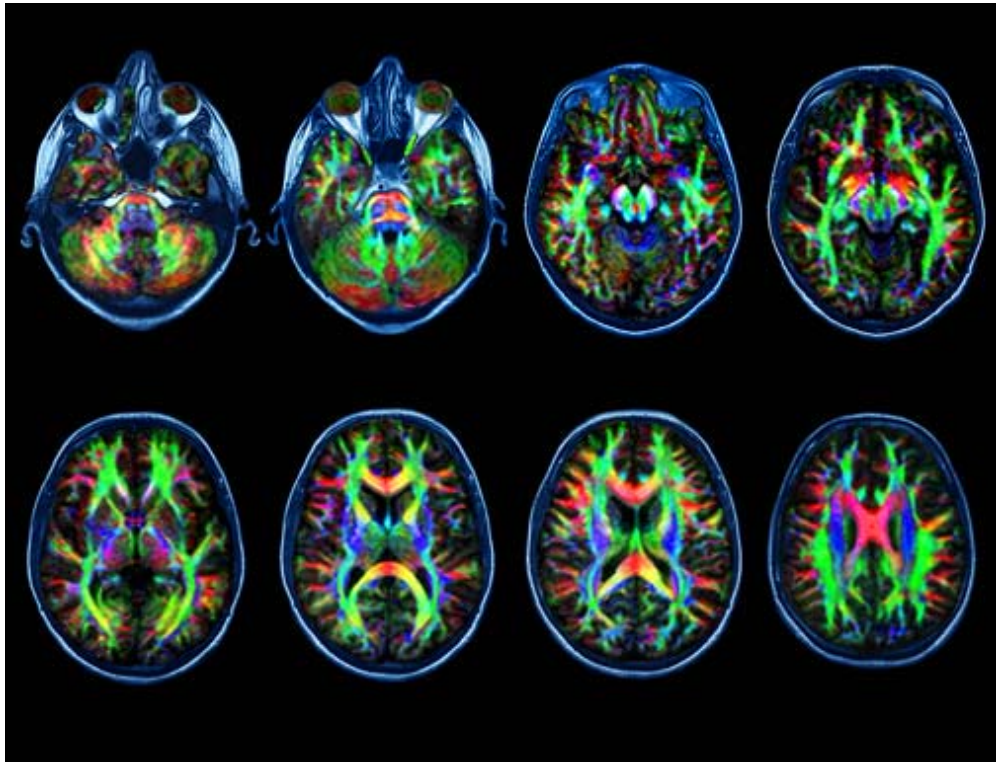
statOmics, Ghent University (<https://statomics.github.io>)

Contents

1	Motivation	2
1.1	Brain imaging study	2
1.2	Challenges	4
1.3	Multiplicity Problem	5
2	False Discovery Rate	9
2.1	Introduction	9
2.2	Intuition of BH-FDR procedure	10
2.3	Benjamini and Hochberg (1995) procedure for controlling the FDR at α	10
2.4	Brain Example	11
2.5	Comments and Extensions	14
3	local fdr	16
3.1	Introduction	16
3.2	Two group model	17
3.3	local fdr	22
3.4	Advantage of having a massive parallel data structure	26
3.5	Power	29
3.6	Comparison with gene expression study	30
	Session info	31

1 Motivation

1.1 Brain imaging study



- Diffusion Tensor Imaging (DTI) data
- DTI measures fluid flows in the brain
- Comparing brain activity of six dyslexic children versus six normal controls
- From each child, DTI produced observations on 15443 voxels (voxel = small volume at a particular (x, y, z) coordinate)
- For each voxel, a two-sided two-sample t-test has been performed, resulting in a z-value (15443 z-values) for fractional anisotropy.
- Low values for FA indicate diffusion in all directions, high values indicates directional diffusion.
- Research question: at what brain locations (voxels) show dyslexic children a different brain activity as compared to children without dyslexia?

For each voxel separately, this is a simple problem, but the large scale of the problem (15443 simultaneous hypothesis tests) causes the problem of multiplicity.

1.1.1 Data Exploration

The dataset `dti` contains

- Spatial location (x, y, z) of each voxel
- z-statistic for assessing differential brain activity between dyslexic and non-dyslexic children

```
library(tidyverse)
library(locfdr)
library(gganimate)

dti <- read_csv("https://raw.githubusercontent.com/statOmics/HDA2020/data/dti.csv",
                col_types = cols())
```

```
pZ <- dti %>%
  ggplot(
    aes(
      coord.y,
      coord.x,
      color=z.value)
  ) +
  geom_point() +
  scale_colour_gradient2(low = "blue",mid="white",high="red") +
  transition_manual(coord.z) +
  labs(title = "transection z = {frame}") +
  theme_grey()
```

We will now plot the animated graph

pZ

WARNING: The animated graph will only be visible in the HTML output, not in PDF format. If you're reading the PDF version, check online for the animated graph.

We visualised the test-statistic of each test per voxel!

Note, that it is difficult to see structure in the data.

1.1.2 Inference

We can convert the z-statistic in a two-sided p-value for each voxel to assess

H_0 : There is on average no difference in brain activity in voxel xyz between dyslexic and non-dyslexic children

$$\mu_d = \mu_{nd}$$

vs

H_0 : There is on average a difference in brain activity in voxel xyz between dyslexic and non-dyslexic children

$$\mu_d \neq \mu_{nd}$$

Below, we calculate the p-values and a variable zP for which we keep the z-value if it is statistical significant at the 5% level otherwise we set it equal to zP=0.

```
dti <- dti %>%
  mutate(
    p.value = pnorm(abs(z.value),lower=FALSE)*2,
    zP = (p.value < 0.05) * z.value)

pPval <- dti %>%
  ggplot(
    aes(
      coord.y,
      coord.x,
      color=zP)
  ) +
  geom_point() +
  scale_colour_gradient2(low = "blue",mid="white",high="red") +
  transition_manual(coord.z) +
  labs(title = "transection z = {frame}") +
  theme_grey()
```

We will now plot the animated graph

```
pPval
```

It is much more easy to observe patterns of activity.

Note, however that

- Higher average FA ($z > 0$ and $p < 0.05$) in dyslexic children is appearing in spatial patterns in some locations.
- Lower average FA ($z < 0$ and $p > 0.05$) in dyslexic children is scattered throughout the brain.
- Multiple testing problem.
- If there would be no association between brain activity and dyslexia we can expect on average $15443 \times \alpha = 772$ false positive voxels at the 5% level of significance.
- Note, that only 1241 were significant at the 5% significance level, so we can expect that the majority of the returned voxels are false positives.

```
FPexpected <- nrow(dti) * 0.05
Preported <- sum(dti$p.value < 0.05)
```

```
FPexpected
```

```
#> [1] 772.15
```

```
Preported
```

```
#> [1] 1241
```

1.2 Challenges

Large Scale Inference implies

- Many hypothesis to be evaluated

- Huge multiple testing problem
- Many false positives can be expected if we do not correct for multiple testing

Issue is widespread in many disciplines

- genomics
- transcriptomics
- proteomics
- brain imaging
- high throughput single cell technologies
- detection of anomalous events: e.g. credit card fraud
- evaluation of trading rules
- academic performance of schools

1.3 Multiplicity Problem

Suppose only a single hypothesis test is required for answering the research question. A statistical test controls the probability of making a **type I error** (type I error rate),

$$\alpha = P[\text{reject } H_0 \mid H_0].$$

The type I error is also known as a **false positive** (i.e. H_0 expresses an negative result, and H_1 a positive result): $\alpha = P[\text{false positive}]$.

An important property:

When H_0 is true, and the assumptions underlying the test hold true, then

$$P \sim U[0, 1].$$

Hence, for any $0 < \alpha < 1$,

$$P[\text{reject } H_0 \mid H_0] = P[P < \alpha \mid H_0] = \alpha.$$

The distribution of the z-statistic and the p-values under H_0 are illustrated below:

```
library(gridExtra)

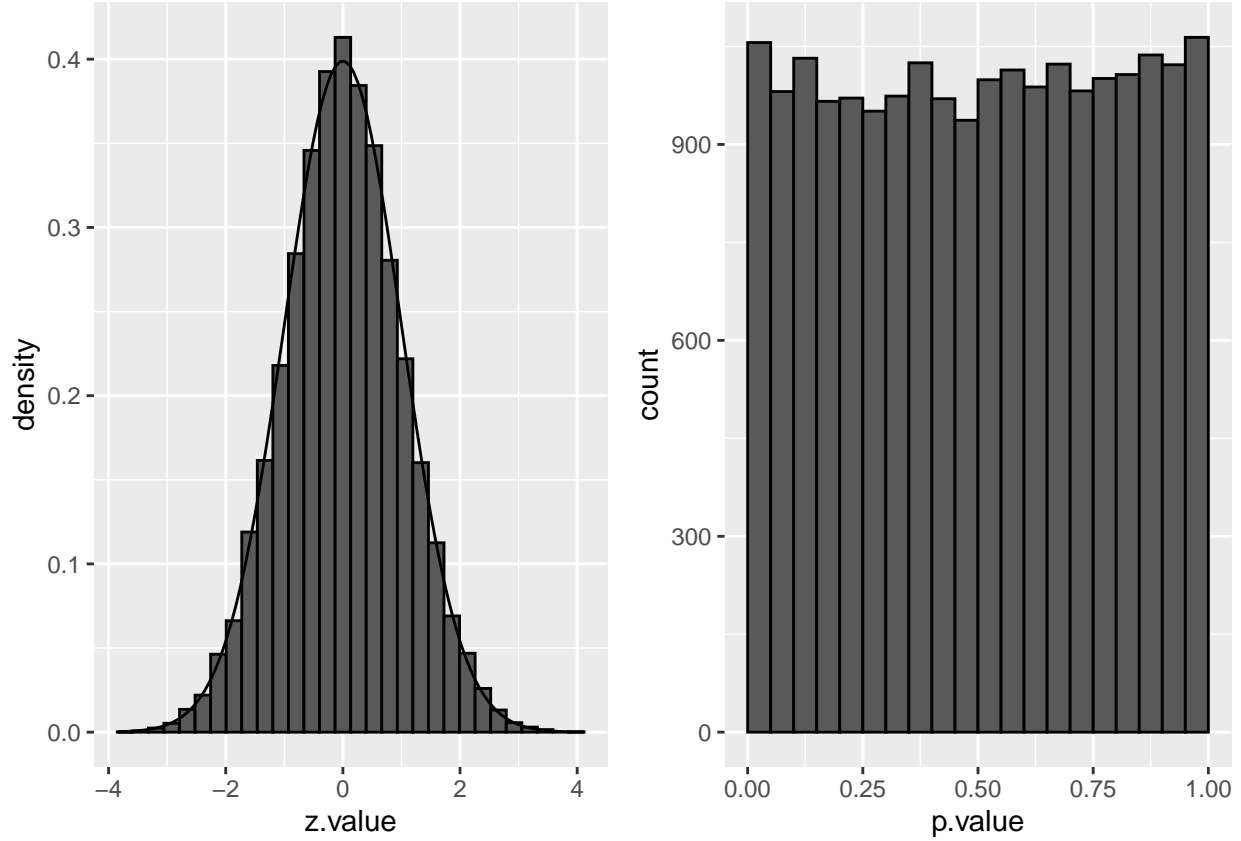
simData <- tibble(
  z.value = rnorm(20000)
)

simData <- simData %>% mutate(p.value = 2*(1-pnorm(abs(z.value))))

p1 <- simData %>%
  ggplot(aes(x = z.value)) +
  geom_histogram(
    aes(y=..density..),
    color = "black") +
  stat_function(fun = dnorm, args=list(mean=0, sd=1))

p2 <- simData %>%
  ggplot(aes(x = p.value)) +
  geom_histogram(color = "black", breaks = seq(0,1,.05))

grid.arrange(p1, p2, ncol=2)
```



We indeed observe that the p-values are uniform under the null hypothesis. So statistical hypothesis testing provides a uniform testing strategy.

1.3.1 Notation

In the multiple testing literature the number of features for which a test is conducted is denoted by m instead of p to avoid confusion with the symbol for a p-value.

Consider testing all $m = 15443$ voxels simultaneously

- What if we assess each individual test at level α ? \rightarrow Probability to have a false positive (FP) among all m simultaneous tests $\gg \alpha = 0.05$
- Indeed for each non differential voxel we have a probability of 5% to return a FP.
- In a typical experiment the majority of the voxels are non differential.
- So an upperbound of the expected FP is $m \times \alpha$ or $15443 \times 0.05 = 772$.

\rightarrow Hence, we are bound to call many false positive voxels each time we run the experiment.

1.3.2 Familywise error rate

Suppose that m hypotheses have to be tested simultaneously for answering a single research question.

Let H_{0i} denote the i th null hypothesis ($i = 1, \dots, m$) and let H_0 denote the intersection of all these partial null hypotheses.

In this case the type I error rate is no longer relevant. Instead one may consider the **Familywise Error Rate (FWER)**

$$\text{FWER} = P[\text{reject at least one } H_{0i} \mid H_0].$$

Assuming independence among the m tests and assuming that all individual tests are performed at the α level of significance, the FWER can be computed as

$$\begin{aligned} \text{FWER} &= P[\text{reject at least one } H_{0i} \mid H_0] \\ &= 1 - P[\text{reject no } H_{0i} \mid H_0] \\ &= 1 - P[\text{not reject } H_{01} \text{ and } \dots \text{ and not reject } H_{0m} \mid H_0] \\ &= 1 - \prod_{i=1}^m P[\text{not reject } H_{0i} \mid H_0] \\ &= 1 - (1 - \alpha)^m. \end{aligned}$$

Examples:

$\alpha = 0.05$ and $m = 5$: FWER= 0.23

$\alpha = 0.05$ and $m = 100$: FWER= 0.99

$\alpha = 0.05$ and $m = 15443$: FWER ≈ 1 .

These calculations illustrate the problem of multiplicity: the more tests that are performed, the larger the probability that at least one false positive conclusion is obtained. Thus if all significant results are listed, and suppose that all null hypotheses hold true, then the FWER is the probability that at least one of the listed positive results is a false positive. Sometimes, a list of significant results represent the “discoveries” from the study, and therefore a false positive result is often also referred to as a false discovery.

For example, with $m = 100$ and $\alpha = 0.05$ the chance that at least one of the “discoveries” is false, is about 99%. Even worse, with $m \approx 15000$ the FWER increases to virtually 100%. In general we also expect that lists of significant results (discoveries) get longer with increasing m .

Many researchers, however, when presented a long list of significant results (or discoveries), would not mind too much if one or a few false discoveries appear in the list. Hence, the FWER is not the most relevant risk measure, as the FWER is allowed to be 100% in case researchers do not mind to have a few false discoveries among the (perhaps many) positive results in the list of discoveries. A better solution will be given later, but first we continue with the use of FWER.

1.3.3 Method of Sidàk: invert FWER to significant level for individual test

The identity $\text{FWER} = 1 - (1 - \alpha)^m$ may be inverted to find the significance level at which each individual test should be tested to attain the nominal familywise error rate at FWER,

$$\alpha = 1 - (1 - \text{FWER})^{1/m}$$

so that the simultaneous testing procedure controls the FWER at the desired level (method of Sidàk).

Examples:

FWER= 0.05 and $m = 5$: $\alpha = 0.0102$

FWER= 0.05 and $m = 100$: $\alpha = 0.00051$

FWER= 0.05 and $m = 15443$: $\alpha = 0.0000033$.

We will argue that this procedure is too stringent for large m .

1.3.4 Bonferroni method

The Bonferroni method is another method that is widely used to control the FWER:

- assess each test at

$$\alpha_{\text{adj}} = \frac{\alpha}{m}$$

- The method does not assume independence of the test statistics.
 - Again, the method is very conservative!
-

To attain the familywise error rate at level FWER the individual hypotheses should be tested at very stringent significance levels when m is large. The consequence of testing at a small significance level α is that it is hard to find significant results, and thus the lists of significant results (discoveries) is likely to be short. Controlling the FWER means that the chance is small that these lists contain one or more false positives. A negative consequence, however, is that many of the true positive hypothesis (i.e. H_1 is true) will not appear in these short lists. Hence, the “power” is small (power is not well defined in this multiple testing setting – extensions of the concept are possible). Thus, avoiding false positives by controlling the FWER comes at a price: many of the true positive hypothesis may be missed.

1.3.5 Adjusted p-value

First we give a very general definition of an **adjusted p-value**.

Define the adjusted p -value as

$$\tilde{p}_i = \{\inf \alpha \in [0, 1] : \text{reject } H_{0i} \text{ at FWER } \alpha\}.$$

With these adjusted p -value, the i th partial null hypothesis may be rejected when

$$\tilde{p}_i < \alpha$$

while controlling the FWER at α .

The corrected p -value should be reported. It accounts for the multiplicity problem and it can be compared directly to the nominal FWER level to make calls at the FWER level.

- adjusted p -values for Bonferroni method:

$$p_{\text{adj}} = \min(p \times m, 1)$$

2 False Discovery Rate

2.1 Introduction

In large scale inference it would be more interesting to tolerate a few false positives as long as they do not dominate the toplist

We first introduce some notation:

The table shows the results of m hypothesis tests in a single experiment.

	accept H_{0i}	reject H_{0i}	Total
null	TN	FP	m_0
non-null	FN	TP	m_1
Total	NR	R	m

- TN : number of true negative: random and unobserved
- FP : number of false positives: random and unobserved
- FN : number of false negatives: random and unobserved
- TP : number of true positives: random and unobserved
- NR : number of acceptances (negative results): random and observed
- R : number of rejections (positive results): random and observed
- m_0 and m_1 : fixed and unobserved
- m : fixed and observed

-
- Note that the table is not completely observable.
 - Indeed, we can only observe the bottom row!
 - The table is introduced to better understand the concept of FWER and to introduce the concept of the false discovery rate (FDR).
-

	accept H_{0i}	reject H_{0i}	Total
null	TN	FP	m_0
non-null	FN	TP	m_1
Total	NR	R	m

The FWER can now be reexpressed as

$$\text{FWER} = \mathbb{P}[\text{reject at least one } H_{0i} \mid H_0] = \mathbb{P}[FP > 0].$$

- However, we know that the FWER is very conservative in large scale inference problems.
- Therefore it would be more interesting to tolerate a few false positives as long as they do not dominate the toplist

The **False Discovery Proportion (FDP)** is the fraction of false positives that are returned, i.e.

$$FDP = \frac{FP}{R}$$

- However, this quantity cannot be observed because in practice we only know the number of voxels for which we rejected H_0 , R .
- But, we do not know the number of false positives, FP .

Therefore, Benjamini and Hochberg, 1995, defined The **False Discovery Rate (FDR)** as

$$\text{FDR} = \text{E} \left[\frac{FP}{R} \right] = \text{E} [\text{FDP}]$$

the expected FDP, in their seminal paper Benjamini, Y. and Hochberg, Y. (1995). “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. Journal of the Royal Statistical Society Series B, 57 (1): 289–300.

- An FDR of 1% means that on average we expect 1% false positive voxels in the list of voxels that are called significant.
- Controlling the FDR allows for more discoveries (i.e. longer lists with significant results), while the fraction of false discoveries among the significant results is well controlled on average. As a consequence, more of the true positive hypotheses will be detected.

2.2 Intuition of BH-FDR procedure

Consider $m = 1000$ voxels

- Suppose that a researcher rejects all null hypotheses for which $p < 0.01$.
- If we use $p < 0.01$, we expect $0.01 \times m_0$ tests to return false positives.
- A conservative estimate of the number of false positives that we can expect can be obtained by considering that the null hypotheses are true for all features, $m_0 = m = 1000$.
- We then would expect $0.01 \times 1000 = 10$ false positives ($FP = 10$).
- Suppose that the researcher found 200 voxels with $p < 0.01$ ($R = 200$).
- The proportion of false positive results ($\text{FDP} = \text{false positive proportion}$) among the list of $R = 200$ genes can then be estimated as

$$\widehat{\text{FDP}} = \frac{FP}{R} = \frac{10}{200} = \frac{0.01 \times 1000}{200} = 0.05.$$

2.3 Benjamini and Hochberg (1995) procedure for controlling the FDR at α

1. Let $p_{(1)} \leq \dots \leq p_{(m)}$ denote the ordered p -values.
2. Find the largest integer k so that

$$\frac{p_{(k)} \times m}{k} \leq \alpha$$

or

$$p_{(k)} \leq k \times \alpha / m$$

3. If such a k exists, reject the k null hypotheses associated with $p_{(1)}, \dots, p_{(k)}$. Otherwise none of the null hypotheses is rejected.

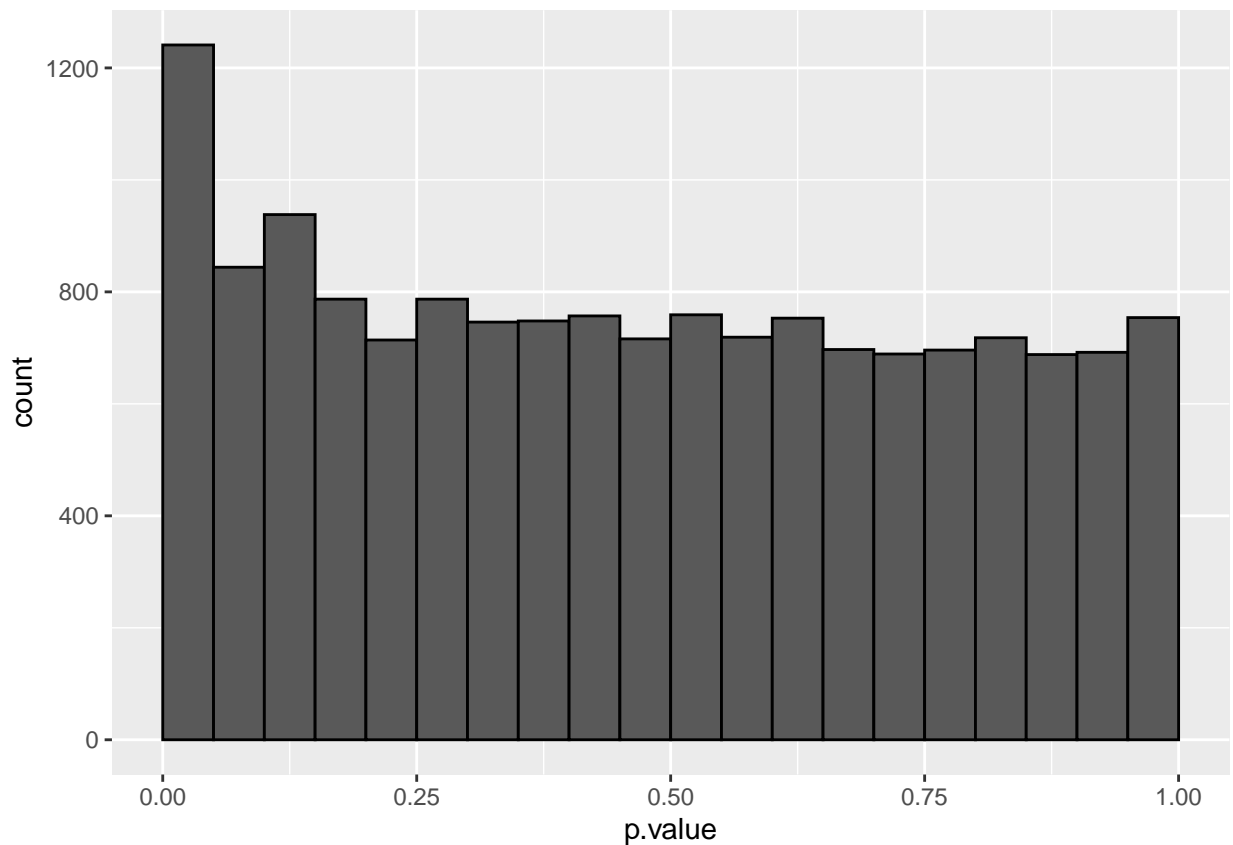
The adjusted p -value (also known as the q -value in FDR literature):

$$q_{(i)} = \tilde{p}_{(i)} = \min \left[\min_{j=i, \dots, m} (mp_{(j)}/j), 1 \right].$$

In the hypothetical example above: $k = 200$, $p_{(k)} = 0.01$, $m = 1000$ and $\alpha = 0.05$.

2.4 Brain Example

```
dti %>%  
  ggplot(aes(x = p.value)) +  
  geom_histogram(color = "black", breaks = seq(0,1,.05))
```

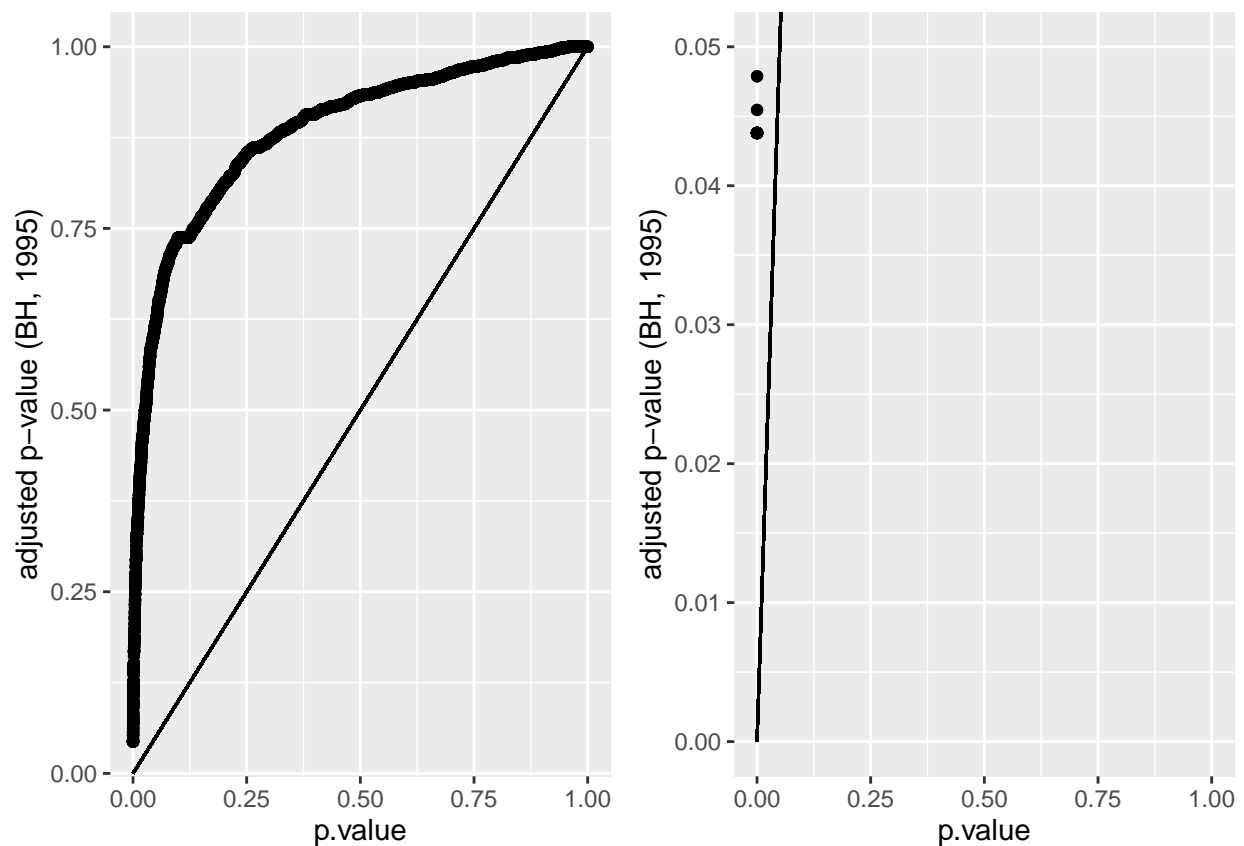


- The graph shows the histogram of the $m = 15443$ p -values. It shows a distribution which is close to a uniform distribution for the larger p -values, but with more small p -values than expected under a uniform distribution.
 - This is a trend that would arise if most of the hypotheses are nulls (resulting in p -values from a uniform distribution), but some are non-nulls (more likely to result in small p -values).
-

```
dti <- dti %>%
  mutate(
    padj = p.adjust(p.value, method="fdr"),
    zFDR = (padj < 0.05) * z.value)

pPadj <- dti %>%
  ggplot(aes(p.value, padj)) +
  geom_point() +
  geom_segment(x=0, y=0, xend=1, yend=1) +
  ylab("adjusted p-value (BH, 1995)")

grid.arrange(pPadj,
  pPadj + ylim(c(0, 0.05)),
  ncol=2)
```



```
# BH corrected p-values
table(dti$padj < 0.05)
```

```
#>
#> FALSE  TRUE
#> 15411   32
```

```
# uncorrected p-values
table(dti$p.value < 0.05)
```

```
#>
#> FALSE TRUE
#> 14202 1241
```

At the 5% FDR, 32 voxels are returned as significantly differentially active between dyslexic and non-dyslexic children.

2.4.1 Ordered table of results to explain the method

- Bonferroni: $\alpha_{\text{adj}} = 3.2e - 06 \rightarrow 0$ voxels are significant at the Bonferroni FWER
- BH-FDR:

1. ordered p -values.
2. Find the largest integer k so that

$$\frac{p_{(k)} \times m}{k} \leq \alpha$$

or

$$p_{(k)} \leq k \times \alpha / m$$

3. If such a k exists, reject the k null hypotheses associated with $p_{(1)}, \dots, p_{(k)}$. Otherwise none of the null hypotheses is rejected.

z.value	p.value	padj	padjNonMonoForm	padjNonMono	adjAlphaForm	adjAlpha	pval < adjAlpha
4.399743	1.08e-05	0.0437969	15443 x pval /1	0.1673701	1 x 0.05/15443	3.20e-06	FALSE
4.336268	1.45e-05	0.0437969	15443 x pval /2	0.1119018	2 x 0.05/15443	6.50e-06	FALSE
4.322818	1.54e-05	0.0437969	15443 x pval /3	0.0792992	3 x 0.05/15443	9.70e-06	FALSE
4.290481	1.78e-05	0.0437969	15443 x pval /4	0.0688318	4 x 0.05/15443	1.30e-05	FALSE
4.273432	1.92e-05	0.0437969	15443 x pval /5	0.0594516	5 x 0.05/15443	1.62e-05	FALSE
4.211374	2.54e-05	0.0437969	15443 x pval /6	0.0653297	6 x 0.05/15443	1.94e-05	FALSE
4.200357	2.66e-05	0.0437969	15443 x pval /7	0.0587925	7 x 0.05/15443	2.27e-05	FALSE
4.116663	3.84e-05	0.0437969	15443 x pval /8	0.0742030	8 x 0.05/15443	2.59e-05	FALSE
4.100929	4.11e-05	0.0437969	15443 x pval /9	0.0706081	9 x 0.05/15443	2.91e-05	FALSE
4.093178	4.26e-05	0.0437969	15443 x pval /10	0.0657102	10 x 0.05/15443	3.24e-05	FALSE
z.value	p.value	padj	padjNonMonoForm	padjNonMono	adjAlphaForm	adjAlpha	pval < adjAlpha
4.077959	4.54e-05	0.0437969	15443 x pval /11	0.0637835	11 x 0.05/15443	3.56e-05	FALSE
4.077082	4.56e-05	0.0437969	15443 x pval /12	0.0586891	12 x 0.05/15443	3.89e-05	FALSE
4.062125	4.86e-05	0.0437969	15443 x pval /13	0.0577663	13 x 0.05/15443	4.21e-05	FALSE
4.047767	5.17e-05	0.0437969	15443 x pval /14	0.0570383	14 x 0.05/15443	4.53e-05	FALSE
4.034977	5.46e-05	0.0437969	15443 x pval /15	0.0562204	15 x 0.05/15443	4.86e-05	FALSE
4.017347	5.89e-05	0.0437969	15443 x pval /16	0.0568081	16 x 0.05/15443	5.18e-05	FALSE
4.010879	6.05e-05	0.0437969	15443 x pval /17	0.0549527	17 x 0.05/15443	5.50e-05	FALSE
4.009839	6.08e-05	0.0437969	15443 x pval /18	0.0521289	18 x 0.05/15443	5.83e-05	FALSE
-4.000404	6.32e-05	0.0437969	15443 x pval /19	0.0513964	19 x 0.05/15443	6.15e-05	FALSE
-4.000404	6.32e-05	0.0437969	15443 x pval /20	0.0488265	20 x 0.05/15443	6.48e-05	TRUE

z.value	p.value	padj	padjNonMonoForm	padjNonMono	adjAlphaForm	adjAlpha	pval < adjAlpha
3.992576	6.54e-05	0.0437969	15443 x pval /21	0.0480641	21 x 0.05/15443	6.80e-05	TRUE
3.977098	6.98e-05	0.0437969	15443 x pval /22	0.0489694	22 x 0.05/15443	7.12e-05	TRUE
3.969507	7.20e-05	0.0437969	15443 x pval /23	0.0483578	23 x 0.05/15443	7.45e-05	TRUE
3.954553	7.67e-05	0.0437969	15443 x pval /24	0.0493390	24 x 0.05/15443	7.77e-05	TRUE
3.950404	7.80e-05	0.0437969	15443 x pval /25	0.0481941	25 x 0.05/15443	8.09e-05	TRUE
3.947772	7.89e-05	0.0437969	15443 x pval /26	0.0468528	26 x 0.05/15443	8.42e-05	TRUE
3.947240	7.91e-05	0.0437969	15443 x pval /27	0.0452178	27 x 0.05/15443	8.74e-05	TRUE
3.946177	7.94e-05	0.0437969	15443 x pval /28	0.0437969	28 x 0.05/15443	9.07e-05	TRUE
3.923751	8.72e-05	0.0454537	15443 x pval /29	0.0464252	29 x 0.05/15443	9.39e-05	TRUE
3.920680	8.83e-05	0.0454537	15443 x pval /30	0.0454537	30 x 0.05/15443	9.71e-05	TRUE

z.value	p.value	padj	padjNonMonoForm	padjNonMono	adjAlphaForm	adjAlpha	pval < adjAlpha
3.899404	0.0000964	0.0478784	15443 x pval /31	0.0480376	31 x 0.05/15443	0.0001004	TRUE
3.892514	0.0000992	0.0478784	15443 x pval /32	0.0478784	32 x 0.05/15443	0.0001036	TRUE
3.863097	0.0001120	0.0523932	15443 x pval /33	0.0523932	33 x 0.05/15443	0.0001068	FALSE
3.847472	0.0001193	0.0527813	15443 x pval /34	0.0542062	34 x 0.05/15443	0.0001101	FALSE
3.846897	0.0001196	0.0527813	15443 x pval /35	0.0527813	35 x 0.05/15443	0.0001133	FALSE

... | ... | ... | ... | ... | ... | ... | ... | ... |

z.value	p.value	padj	padjNonMonoForm	padjNonMono	adjAlphaForm	adjAlpha	pval < adjAlpha
0.0003165	0.9997475	0.9999417	15443 x pval /15440	0.9999417	15440 x 0.05/15443	0.0499903	FALSE
-0.0002325	0.9998145	0.9999440	15443 x pval /15441	0.9999440	15441 x 0.05/15443	0.0499935	FALSE
-0.0000953	0.9999240	0.9999665	15443 x pval /15442	0.9999887	15442 x 0.05/15443	0.0499968	FALSE
0.0000420	0.9999665	0.9999665	15443 x pval /15443	0.9999665	15443 x 0.05/15443	0.0500000	FALSE

```
pFDR <- dti %>%
  ggplot(
    aes(
      coord.y,
      coord.x,
      color=zFDR)
  ) +
  geom_point() +
  scale_colour_gradient2(low = "blue",mid="white",high="red") +
  transition_manual(coord.z) +
  labs(title = "transection z = {frame}") +
  theme_grey()
```

2.4.2 Visualisation of significant differences in brain activity at the 5% FDR

2.5 Comments and Extensions

- Benjamini and Hochberg published their method in 1995; it was one of the first FDR control methods.
- The same authors published later yet other FDR control methods.
- For this reason their 1995 method is often referred to as the Benjamini and Hochberg 1995 method, or BH95.
- As input the method only needs the p -values from the m hypotheses tests.
- When controlling FDR, the adjusted p -values are often referred to as q -values.

-
- It is a **linear step-up procedure** : it starts from the least significant result (largest p-value) and steps-up to more significant results (lower p-values).
 - In FDR terminology the adjusted p -value is often referred to as a q -value.
 - The BH95 method assumes that all tests are mutually independent (or at least a particular form of positive dependence between the p -values).
 - When the assumptions hold, it guarantees

$$\text{FDR} = \mathbb{E}[TP/R] = \mathbb{E}[\text{FDP}] \leq \frac{m_0}{m} \alpha \leq \alpha.$$

2.5.1 Extension

Thus, if we knew m_0 (the number of true nulls), we could improve the method by applying it to the level $\alpha m/m_0$ (cfr. Bonferroni).

→ many FDR methods consist in estimating m_0 or the fraction of null genes m_0/m .

The inequality

$$\text{FDR} \leq \frac{m_0}{m} \alpha \leq \alpha$$

shows that BH1995 is a conservative method, i.e. it controls the FDR at the safe side, i.e. when one is prepared to control the FDR at the nominal level α , the BH95 will guarantee that the true FDR is not larger than the nominal level (when the assumptions hold).

- More interestingly is that $\frac{m_0}{m} \alpha$ is in between the true FDR and the nominal FDR.
- Suppose that m_0 were known and that the BH95 method were applied at the nominal FDR level of $\alpha = m/m_0 \alpha^*$, in which α^* is the FDR level we want to control. Then the inequality gives

$$\text{FDR} \leq \frac{m_0}{m} \alpha = \frac{m_0}{m} \frac{m}{m_0} \alpha^* = \alpha^*,$$

and hence BH95 would better control the FDR at α^* .

- Note that $\alpha = m/m_0 \alpha^* > \alpha^*$ and hence the results is less conservative than the original BH95 method.
-

The above reasoning implies a **generalized adaptive linear step-up procedure**:

- estimate m_0 : \hat{m}_0
- of $\hat{m}_0 = 0$, reject all null hypotheses; otherwise, apply the step-up procedure of BH 95 at the level $\alpha = m \alpha^* / \hat{m}_0$ to control the FDR at α^* .

The adjusted p -values (= q -values) are obtained as

$$\tilde{p}_{(i)} = \frac{\hat{m}_0}{m} \min \left\{ \min_{j=i, \dots, m} \{m p_{(j)} / j\}, 1 \right\}.$$

- Many FDR procedures can be fit into this definition (e.g. Benjamini and Hochberg (2000) and Tibshirani (2003)).
 - We do not give details on the methods for estimating m_0 , but some of them are implemented in the R software. On the next page we illustrate with simulated data that BH can be improved with estimated m_0 .
-

2.5.2 Other important considerations

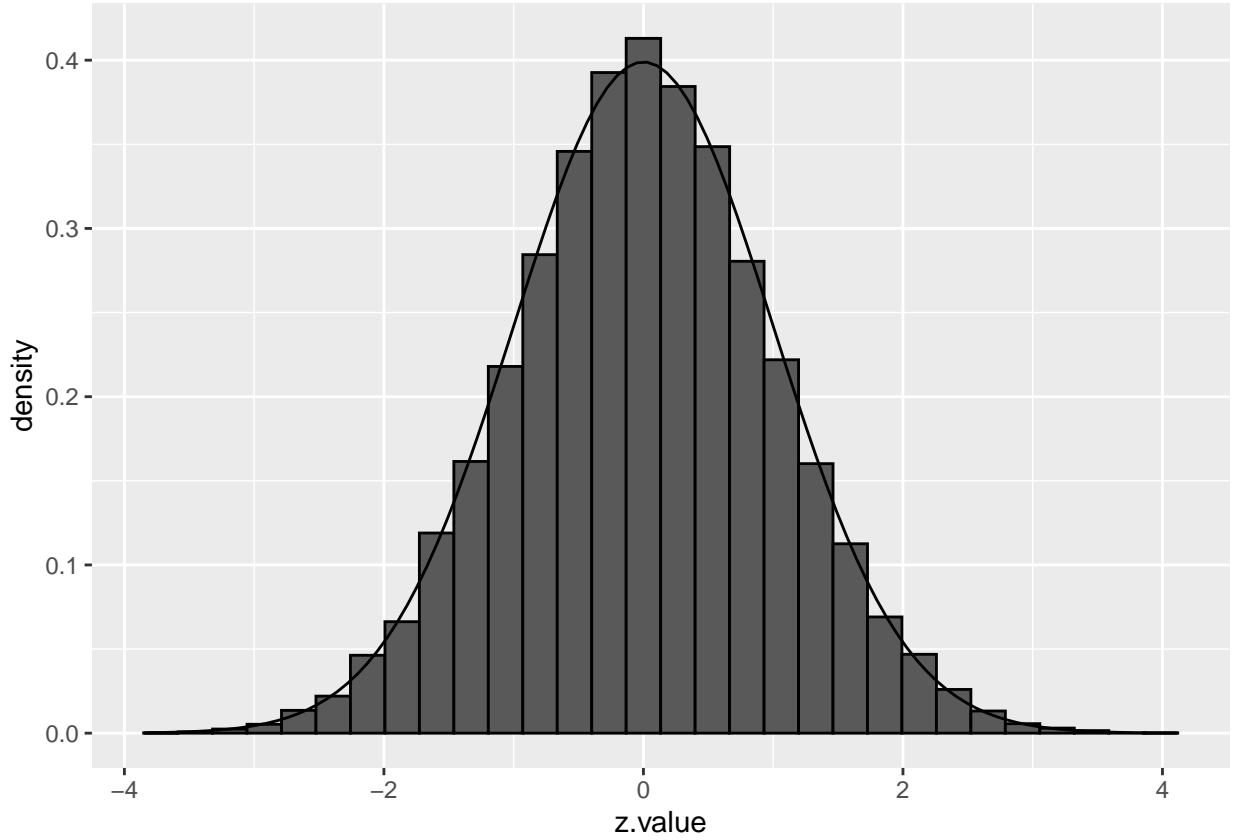
- It can be shown that the BH-FDR method weakly controls the FWER, i.e. it controls the FWER if all features are false ($m_0 = m$).
- The BH-FDR is derived under the assumption of independence of the features and has been shown to be only valid under special forms of dependence between the features.

3 local fdr

3.1 Introduction

Suppose that the test statistic for testing H_{0i} is denoted by z_i , and that the test statistics have a $N(0, 1)$ null distribution.

If all m null hypotheses are true, the histogram of the m test statistics should approximate the theoretical null distribution (density $f_0(z)$).



Assuming that the test statistic has a standard normal null distribution is not restrictive. For example, suppose that t -tests have been applied and that the null distribution is t_d , with d representing the degrees of freedom. Let F_{td} denote the distribution function of t_d and let Φ denote the distribution function of the standard normal distribution. If T denotes the t -test statistic, then, under the null hypothesis,

$$T \sim t_d$$

and hence

$$F_{td}(T) \sim U[0, 1]$$

and

$$Z = \Phi^{-1}(F_{td}(T)) \sim N(0, 1).$$

If all m null hypotheses are true, then each of the Z_i is $N(0, 1)$ and the set of m calculated z_i test statistics may be considered as a sample from $N(0, 1)$. Hence, under these conditions we expect the histogram of the m z_i 's to look like the density of the null distribution.

3.2 Two group model

- Suppose that under the alternative hypothesis the test statistic has density function $f_1(z)$.
- We use the term “null” to refer to a case i for which H_{0i} is true, and “non-null” for a case i for which H_{0i} is not true.
- Consider the **prior probabilities**

$$\pi_0 = P[\text{null}] \text{ and } \pi_1 = P[\text{non-null}] = 1 - \pi_0.$$

- The marginal distribution of the m test statistics is then given by the **mixture distribution**

$$f(z) = \pi_0 f_0(z) + \pi_1 f_1(z)$$

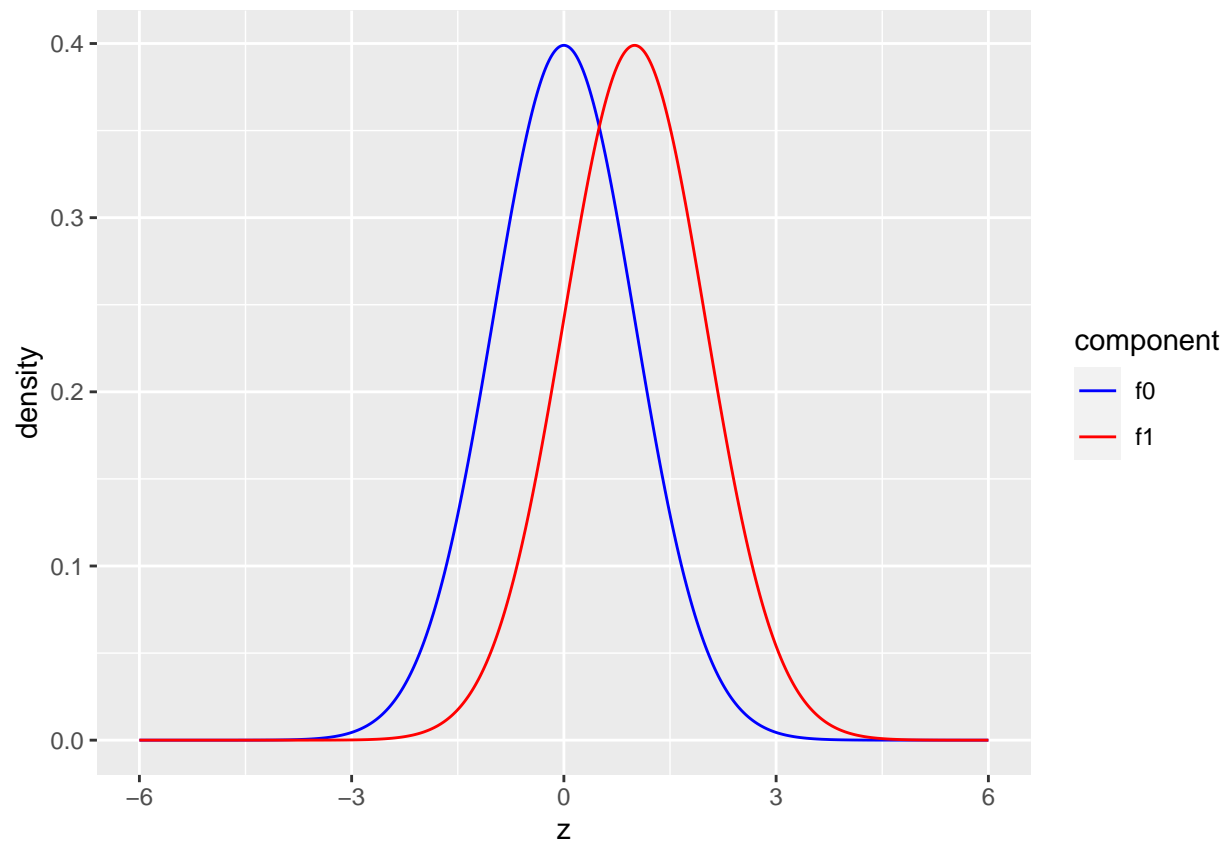
3.2.1 Examples of mixture distributions

We have already explored mixture distributions in detail in the paper reading session on model based clustering.

- blue: f_0 : $N(0, 1)$, red: f_1 : $N(1, 1)$

```
components <- tibble(z = seq(-6,6,.01)) %>%
  mutate(
    f0 = dnorm(z),
    f1 = dnorm(z, mean = 1))

components %>%
  gather(component, density, -z) %>%
  ggplot(aes(z,density,color = component)) +
  geom_line() +
  scale_color_manual(values=c("blue","red"))
```

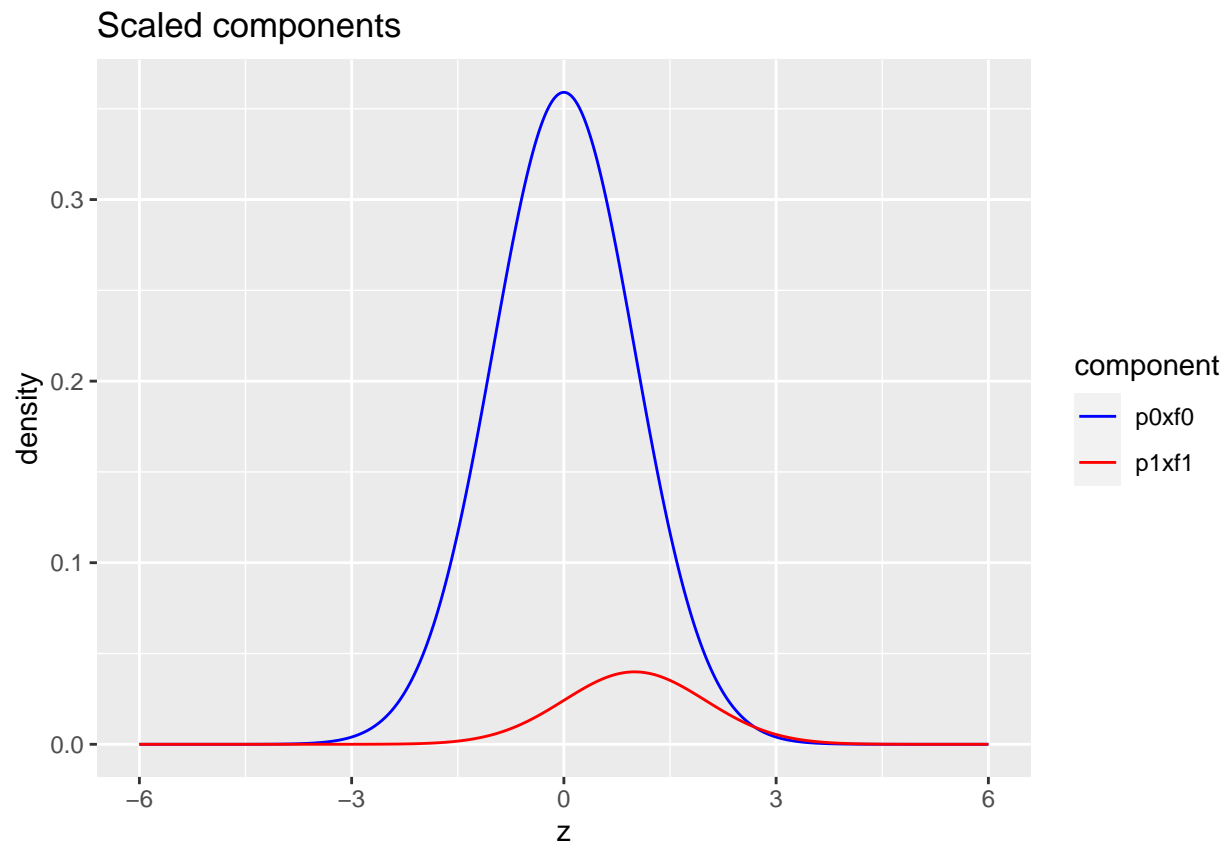


The graphs shows the two component distributions separately.

- blue: $\pi_0 \times f_0$ with $\pi_0 = 0.9$ and $f_0 = N(0, 1)$
- red: $\pi_1 \times f_1$ with $\pi_1 = 1 - \pi_0 = 0.1$ and $f_1 = N(1, 1)$

```
p0 <- 0.9
p1 <- 1-p0
mu1 <- 1
scaledComponents <- tibble(z = seq(-6,6,.01)) %>%
  mutate(
    p0xf0 = dnorm(z) * p0,
    p1xf1 = dnorm(z, mean = mu1)*p1
  )

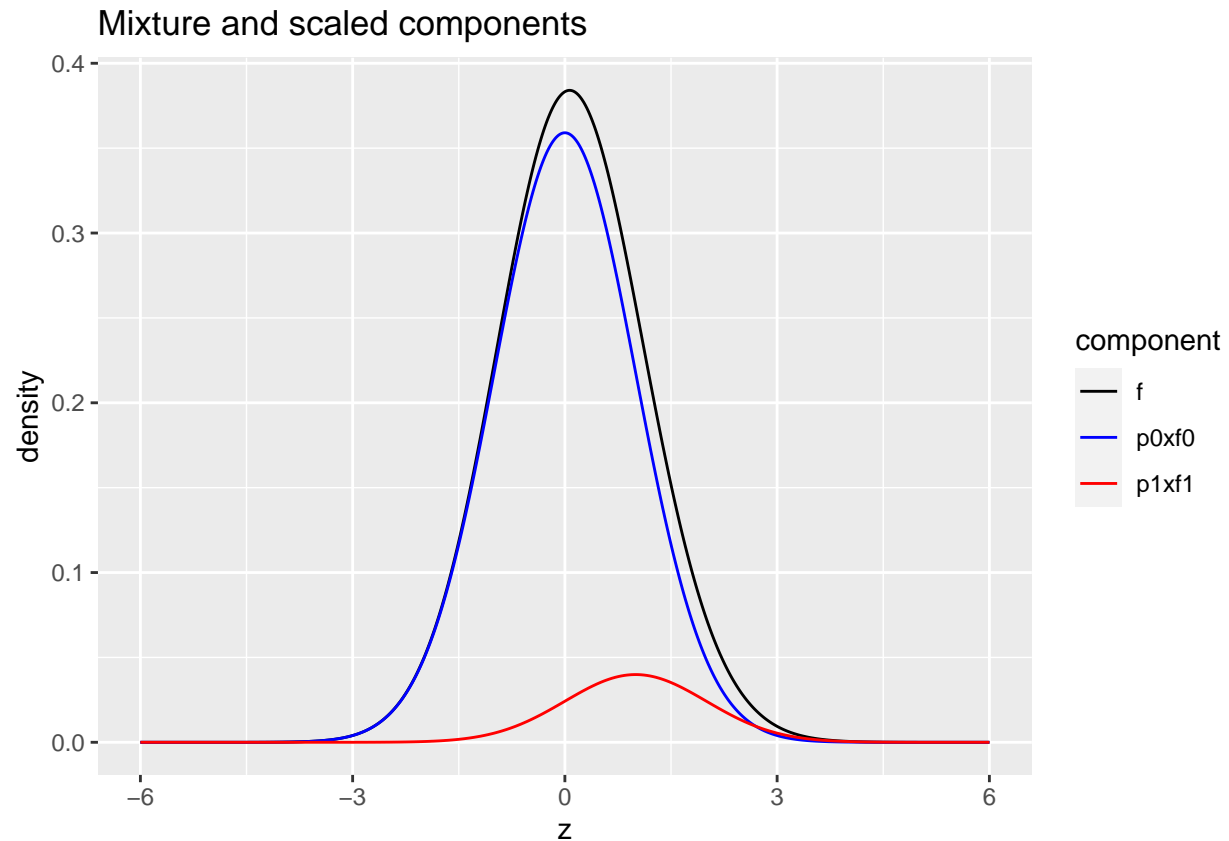
scaledComponents %>%
  gather(component, density, -z) %>%
  ggplot(aes(z,density,color = component)) +
  geom_line() +
  scale_color_manual(values=c("blue","red")) +
  ggtitle("Scaled components")
```



Mixture distribution

- blue: $\pi_0 \times f_0$ with $\pi_0 = 0.9$ and $f_0 = N(0, 1)$
- red: $\pi_1 \times f_1$ with $\pi_1 = 1 - \pi_0 = 0.1$ and $f_1 = N(1, 1)$
- black: $f = \pi_0 f_0 + \pi_1 f_1$

```
scaledComponents %>%
  mutate(f=p0xf0+p1xf1) %>%
  gather(component, density, -z) %>%
  ggplot(aes(z,density,color = component)) +
  geom_line() +
  scale_color_manual(values=c("black","blue","red")) +
  ggtitle("Mixture and scaled components")
```

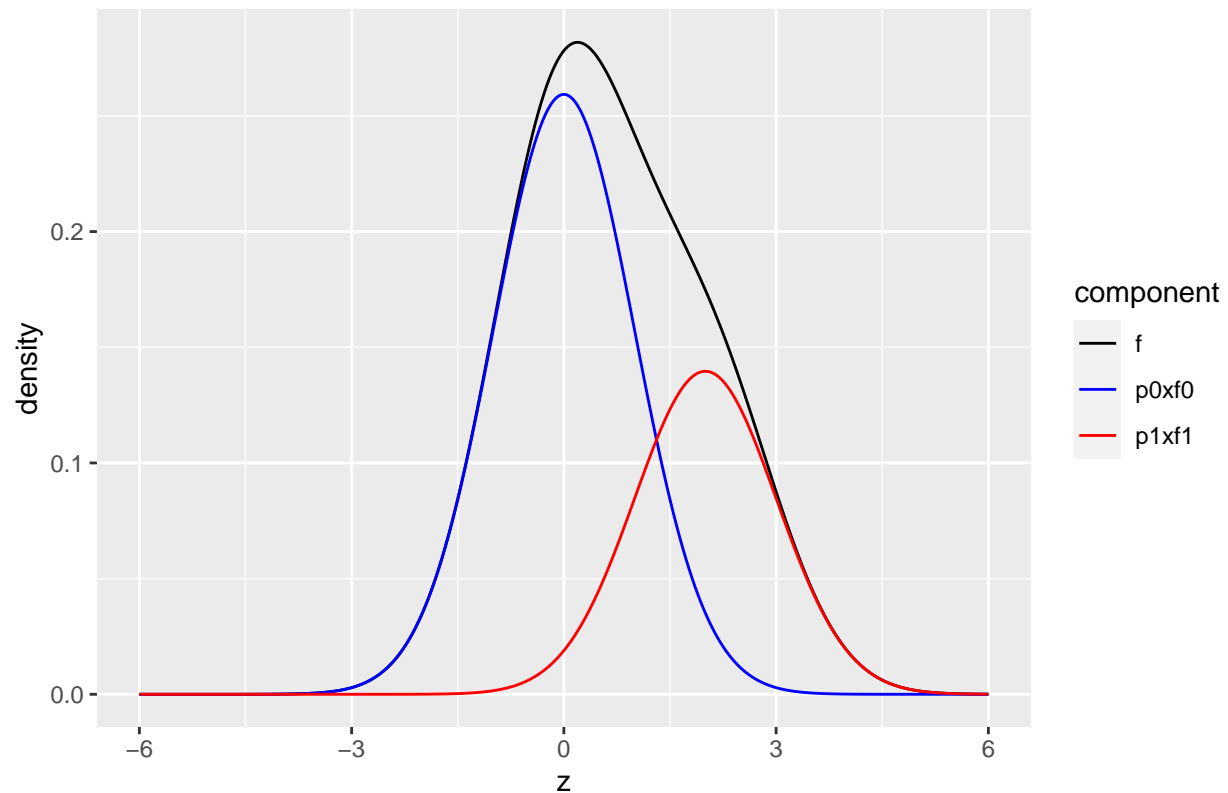


Mixture $\pi_0 f_0(z) + \pi_1 f_1(z)$ with $\pi_0 = 0.65$ and $f_1 = N(2, 1)$ and $f_0 = N(0, 1)$

```
p0 <- 0.65
p1 <- 1-p0
mu1 <- 2
scaledComponents <- tibble(z = seq(-6,6,.01)) %>%
  mutate(
    p0xf0 = dnorm(z) * p0,
    p1xf1 = dnorm(z, mean = mu1)*p1)

scaledComponents %>%
  mutate(f=p0xf0+p1xf1) %>%
  gather(component, density, -z) %>%
  ggplot(aes(z,density,color = component)) +
  geom_line() +
  scale_color_manual(values=c("black","blue","red")) +
  ggtitle("Mixture and scaled components (p0 = 0.35)")
```

Mixture and scaled components ($p_0 = 0.35$)



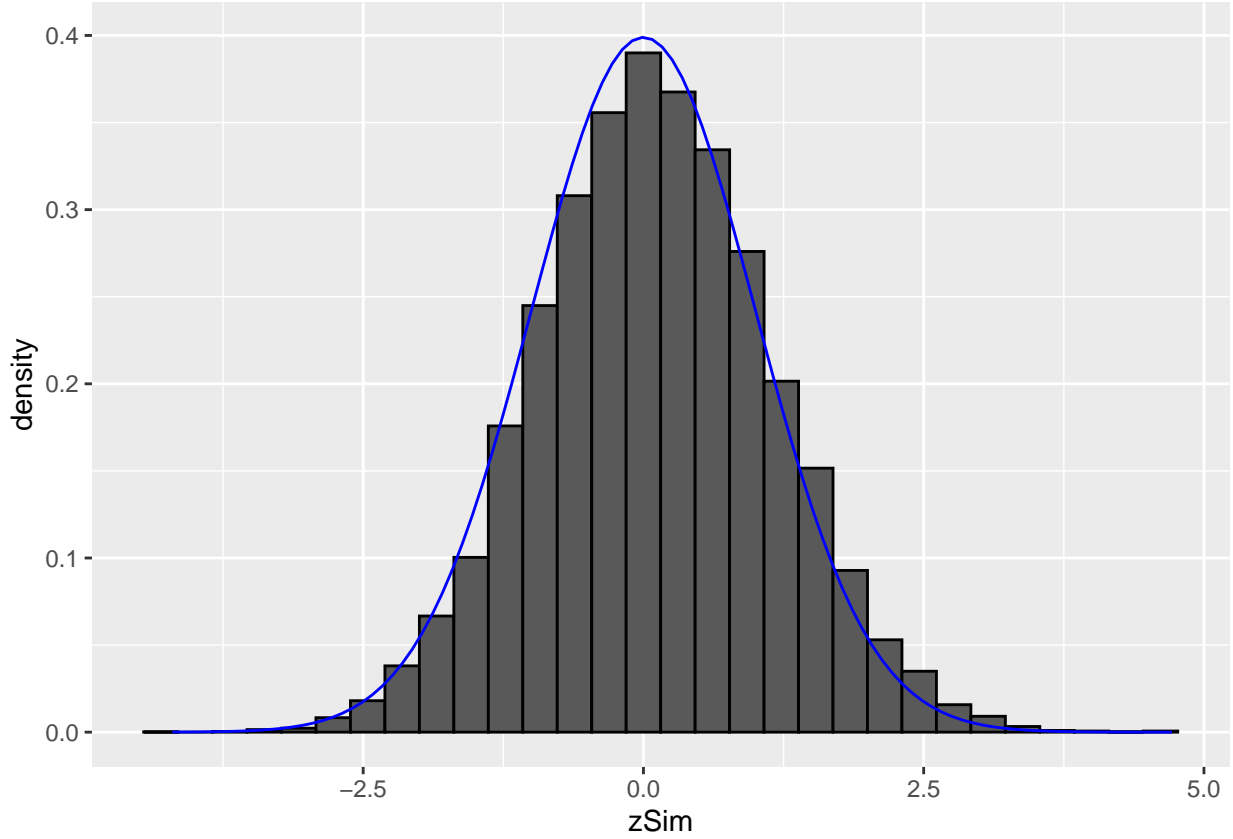
3.2.2 simulations

Simulated data: 20000 z -statistics with $\pi_1 = 0.10$ non-nulls with $f_1 = N(1, 1)$.

```
p0 <- .9
p1 <- 1-p0
mu1 <- 1
m <- 20000

zSim <- c(
  rnorm(m * p0),
  rnorm(m * p1, mean=mu1)
)

zSim %>%
  as_tibble %>%
  ggplot(aes(x = zSim)) +
  geom_histogram(
    aes(y=..density..),
    color = "black") +
  stat_function(fun = dnorm,
    args = list(
      mean = 0,
      sd=1),
    color="blue")
```



It is hard to see the difference between the histogram and the density function of the null distribution (blue curve), because the mean of f_1 is not much larger than 0 and because only $\pi_1 = 10\%$ non-nulls are included and because the alternative is not far from the null distribution. However, this is not an unrealistic setting.

Note, that in most settings the non-null features will originate from a mixture of multiple distributions with positive and negative means. Fortunately, the local fdr method does not require us to estimate f_1 as we will see further.

3.3 local fdr

We can now calculate the probability that a case is a null given the observed z ,

$$P[\text{null} \mid z] = \frac{\pi_0 f_0(z)}{f(z)}.$$

This probability is referred to as the **local false discovery rate**, and denoted by $\text{fdr}(z)$.

If for an observed z , $\text{fdr}(z)$ is sufficiently small, one may believe that the case is a true discovery (i.e. H_{0i} may be rejected).

3.3.1 Link with FDR

Recall the definition of the FDR,

$$\text{FDR} = \mathbb{E}[FP/R] \quad (1)$$

$$= \mathbb{E}[\text{number of nulls among rejected}/\text{number of rejected}] \quad (2)$$

$$= \mathbb{P}[\text{null} \mid \text{rejected}] \quad (3)$$

- The FDR is to be interpreted as an overall risk: *among all rejected hypotheses* (discoveries) it gives the expected fraction (or probability) of a null (false discovery).
- The local fdr, on the other hand, is to be interpreted as a risk for a specific decision: if a null hypothesis is rejected based on a test statistic value of z , then the local fdr gives the probability of that single discovery being a false discovery.
- Since the local fdr has a clear interpretation that applies to an individual hypothesis test, it can be used to decide whether or not to reject a null hypothesis.
- In particular, reject a null hypothesis H_{0i} if $\text{fdr}(z) < \alpha$, where α is the nominal local fdr level at which the multiple testing problem need to be controlled at.
- The local fdr method can only be applied if π_0 and f can be estimated from the data (see later). The density f_0 can be either known (null distribution of the test statistic) or it can be estimated from the observed m test statistics.

For the sake of simplicity, suppose that H_{0i} is tested against a one-sided alternative and that H_{0i} is rejected for small z , i.e.

$$H_0 : z = 0 \text{ vs } H_1 : z < 0$$

Suppose that all H_{0i} are rejected for which the observed test statistic is at most z , then we can write

$$\text{FDR}(z) = \mathbb{P}[\text{null} \mid \text{rejected}] \quad (4)$$

$$= \mathbb{P}[\text{null} \mid Z \leq z] \quad (5)$$

$$= \mathbb{P}[\text{null} \mid Z \leq z] \quad (6)$$

$$= \mathbb{E}_Z \{\mathbb{P}[\text{null} \mid Z] \mid Z \leq z\} \quad (7)$$

$$= \mathbb{E}_Z \{\mathbb{P}[\text{null} \mid Z] \mid Z \leq z\} \quad (8)$$

$$= \mathbb{E}_Z [\text{fdr}(Z) \mid Z \leq z] \quad (9)$$

$$= \mathbb{E}_Z [\text{fdr}(Z) \mid Z \leq z] \quad (10)$$

$$= \mathbb{E}_Z [\text{fdr}(Z) \mid Z \leq z] \quad (11)$$

$$= \frac{\int_{-\infty}^z \text{fdr}(u) f(u) du}{\int_{-\infty}^z f(u) du} \quad (12)$$

$$= \frac{\pi_0 \int_{-\infty}^z f_0(u) du}{F(z)} \quad (13)$$

$$= \frac{\pi_0 \int_{-\infty}^z f_0(u) du}{F(z)} \quad (14)$$

$$= \frac{\pi_0 F_0(z)}{F(z)} \quad (15)$$

$$= \frac{\pi_0 F_0(z)}{F(z)} \quad (16)$$

This shows that $\text{fdr}(z) = \frac{\pi_0 f_0(z)}{f(z)}$ and $\text{FDR}(z) = \frac{\pi_0 F_0(z)}{F(z)}$ have similar expression. The former is expressed in terms of density functions, and the latter in terms of the corresponding cumulative distribution functions.

From the equality

$$\text{FDR}(z) = \frac{\int_{-\infty}^z \text{fdr}(u) f(u) du}{\int_{-\infty}^z f(u) du}$$

we learn that the probability for a false discovery among hypotheses rejected by using threshold z , equals the average of the local false discovery rates $\text{fdr}(u)$ of the discoveries ($u \leq z$ here).

Note, that the BH-FDR adopts

- $\pi_0 = 1$, which is a conservative estimate
- uses the theoretical null for $p = F_0(z)$
- uses the empirical cumulative distribution function $\bar{F}(z) = \frac{\#Z \leq z}{m}$ to estimate $F(z)$.

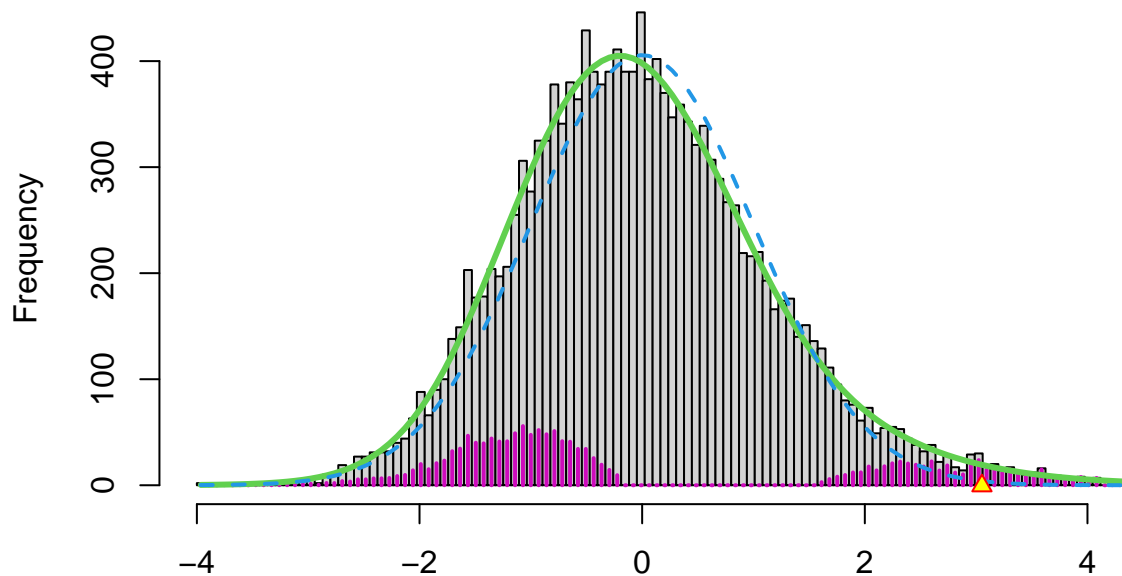
A similar identity can be easily shown for two-sided tests.

3.3.2 Estimation of $\text{fdr}(z) = \frac{\pi_0 f_0(z)}{f(z)}$

- $f(z)$ can be estimated by nonparametric density estimation methods ($f(z)$ is the marginal distribution of the test statistics; no knowledge about null / non-null is needed)
- $f_0(z)$ is known or can be estimated from the data
- π_0 can be estimated once $f(z)$ and $f_0(z)$ are estimated for all z .

3.3.3 Brainscan example

```
library(locfdr)
lfdr <- locfdr(dti$z.value, nulltype = 0)
```

- In the brainscan example the test statistics are supposed to be $N(0,1)$ distributed under the null hypothesis. Tests are performed two-sided.
- The argument `nulltype=0` specifies that the null distribution (f_0) is $N(0,1)$.
- The dashed blue line gives f_0 and the solid green line is the nonparametric estimate of the marginal density function f . The two densities do not coincide and hence we may anticipate that some of the voxels show differential brain activity.
- The purple bars indicate the estimated number of non-nulls (among the hypotheses/voxels for a given z -value). The plots shows that more non-nulls are expected for the negative z -values than for the positive z -values (sign of z corresponds to more or less brain activity in normal versus dyslectic children).

3.3.4 Problems?

Note, however, that

- we typically expect that the majority of the test statistics follow the null distribution.
- that the null distribution in the plot is rescaled
- So, we would expect that the two distributions to overlay in the middle part.
- However, we observe a shift.

In practise it often happens that the theoretical null distribution is not valid.

This can happen due to

1. Failed mathematical assumptions: null distribution is incorrect
2. Correlation between the samples
3. Correlation between the features
4. Confounding that is not corrected for.

3.4 Advantage of having a massive parallel data structure

The massive parallel data structure enables us

- to spot deviations from the theoretical null distribution.
- to estimate the null distribution by using all features.

Efron relaxes the local fdr method by assuming that the null distribution is a Normal distribution but with a mean and variance that can be estimated empirically (based on all the features).

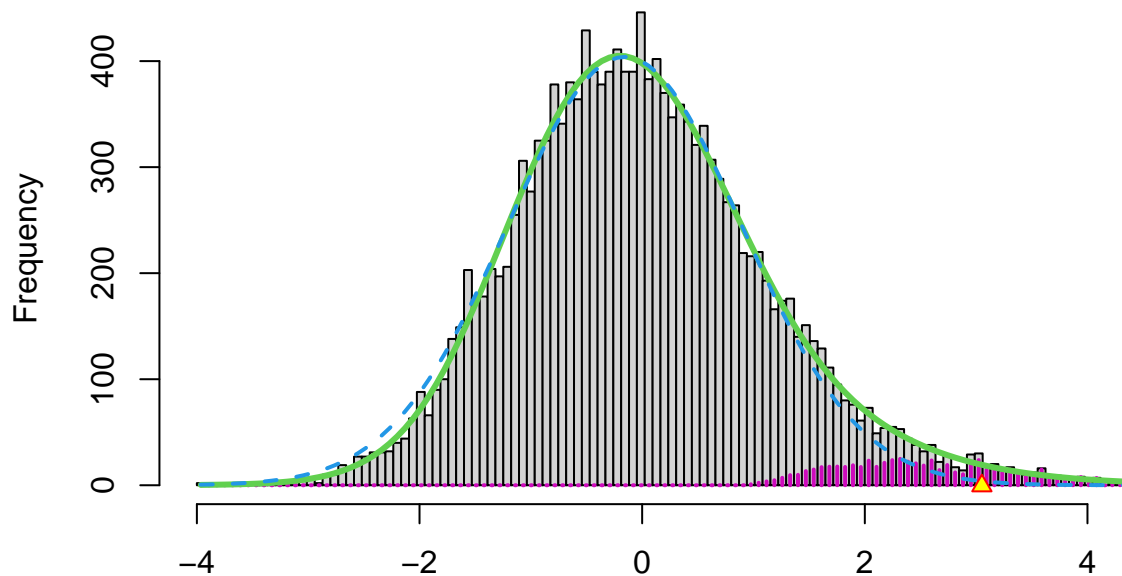
This can be done by setting the argument `nulltype` in the `locfdr` function equal to `nulltype = 1`, which is the default or by setting `nulltype = 2`.

The `locfdr` method then uses

1. `nulltype = 1` maximum likelihood to estimate the null by only considering the middle part in the distribution of the test statistics (MLE) or
2. `nulltype = 2` a geometric method that places the best fitting normal under the peak of the estimate of $f(z)$. (CME)

3.4.1 Brainscan example

```
lfdr <- locfdr(dti$z.value)
```

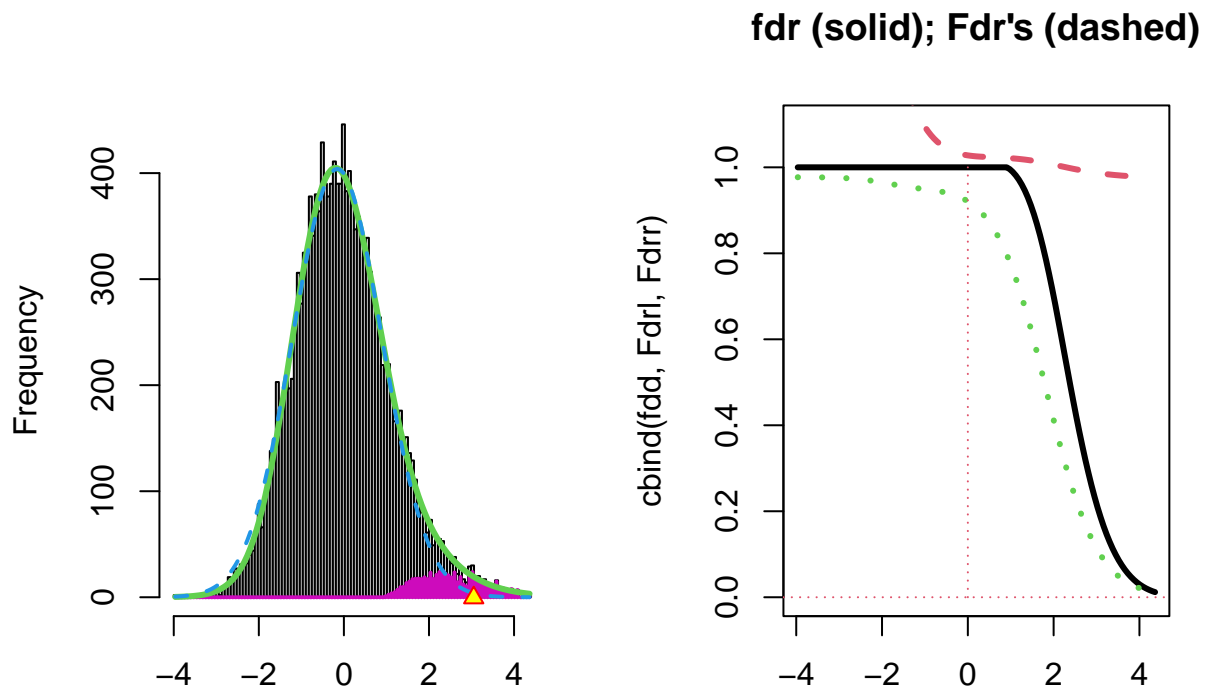


MLE: delta: -0.157 sigma: 1.052 p0: 0.977
 CME: delta: -0.191 sigma: 1.066 p0: 0.992

The plot shows that the null distribution is shifted to negative values and has a standard deviation that remains close to 1.

- This often happens if there is correlation between the features.
- Spatial correlation can be expected in the brain, so voxels that are close to each-other typically will be correlated.
- The dashed blue line gives f_0 and the solid green line is the nonparametric estimate of the marginal density function f . The two densities do not coincide and hence we may anticipate that some of the voxels show differential brain activity.
- The purple bars indicate the estimated number of non-nulls (among the hypotheses/voxels for a given z -value). The plots shows that only non-nulls for positive z -values are expected (sign of z corresponds to more or less brain activity in normal versus dyslectic children).

```
lfdr <- locfdr(dti$z.value, plot=2)
```



MLE: delta: -0.157 sigma: 1.052 p0: 0.97
 CME: delta: -0.191 sigma: 1.066 p0: 0.97

Efdr= 0.468

- The plot at the left is the same as on the previous page.
- The plot at the right shows the local fdr as the black solid line. Close to $z = 0$ the fdr is about 1 (i.e. if those hypotheses would be rejected, the probability of a false positive is about 100%). When moving away from $z = 0$ to larger values the fdr drops.
- This means that we can only discover convincingly differential brain activity for large positive z . Rejecting null hypotheses with large negative z would still be risky: large chance of false discovery.
- The reason can be read from the first graph: for negative z the ratio $f_0(z)/f(z)$ is almost 1, whereas for large positive z the ratio $f_0(z)/f(z)$ becomes small.
- Note, that the result is atypically. In most applications we typically pick-up both downregulated (negative z) and upregulated (positive z) features.

```
dti <- dti %>%
  mutate(
    lfdr = lfdr$fdr,
    zfdr = (lfdr < 0.2) * z.value)

pfdr <- dti %>%
  ggplot(
    aes(
      coord.y,
```

```

    coord.x,
    color=zfdr)
  ) +
  geom_point() +
  scale_colour_gradient2(low = "blue",mid="white",high="red") +
  transition_manual(coord.z) +
  labs(title = "transection z = {frame}") +
  theme_grey()

```

Note, that the local fdr method allows us to detect differential brain activity in a specific region in the front part of the brain for which a larger fractional anisotropy is observed on average for children having dyslexia.

We can also estimate the FDR of the set that we return as the average local fdr in this set.

```

dti %>%
  filter(lfdr < 0.2) %>%
  pull(lfdr) %>%
  mean

```

```
#> [1] 0.1034925
```

3.5 Power

The local false discovery rate may also be used to get **power diagnostics**.

General idea: for z 's supported by the alternative hypothesis (i.e. large $f_1(z)$), we hope to see small $\text{fdr}(z)$.

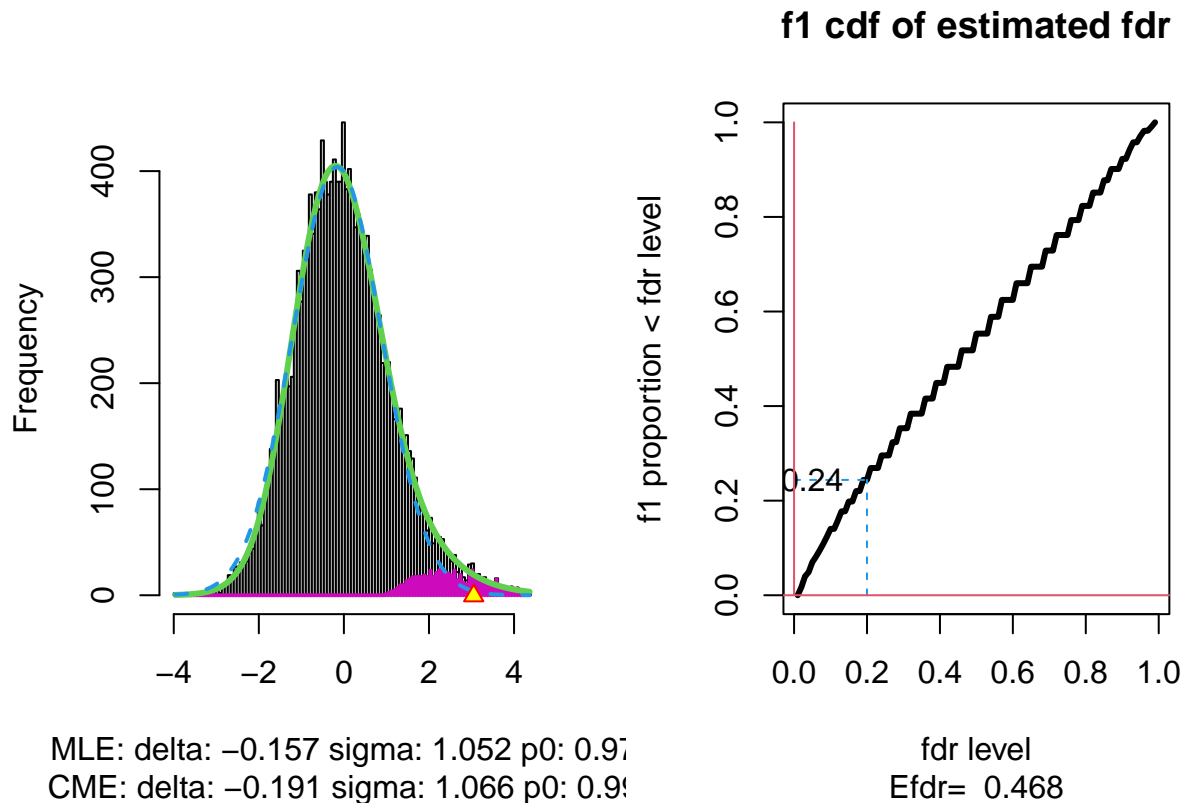
The **expected fdr** is an appropriate summary measure:

$$\text{Efdr} = E_{f_1}[\text{fdr}(Z)] = \int_{-\infty}^{+\infty} \text{fdr}(z) f_1(z) dz.$$

With estimates of $\text{fdr}(z)$ and $f_1(z)$, the Efdr can be computed.

A small Efdr is an indication of a powerful study.

```
lfdr <- locfdr(dti$z.value, plot = 3)
```



With α the nominal local fdr level, the vertical axis gives

$$E_{f_1} [\text{fdr}(Z) < \alpha].$$

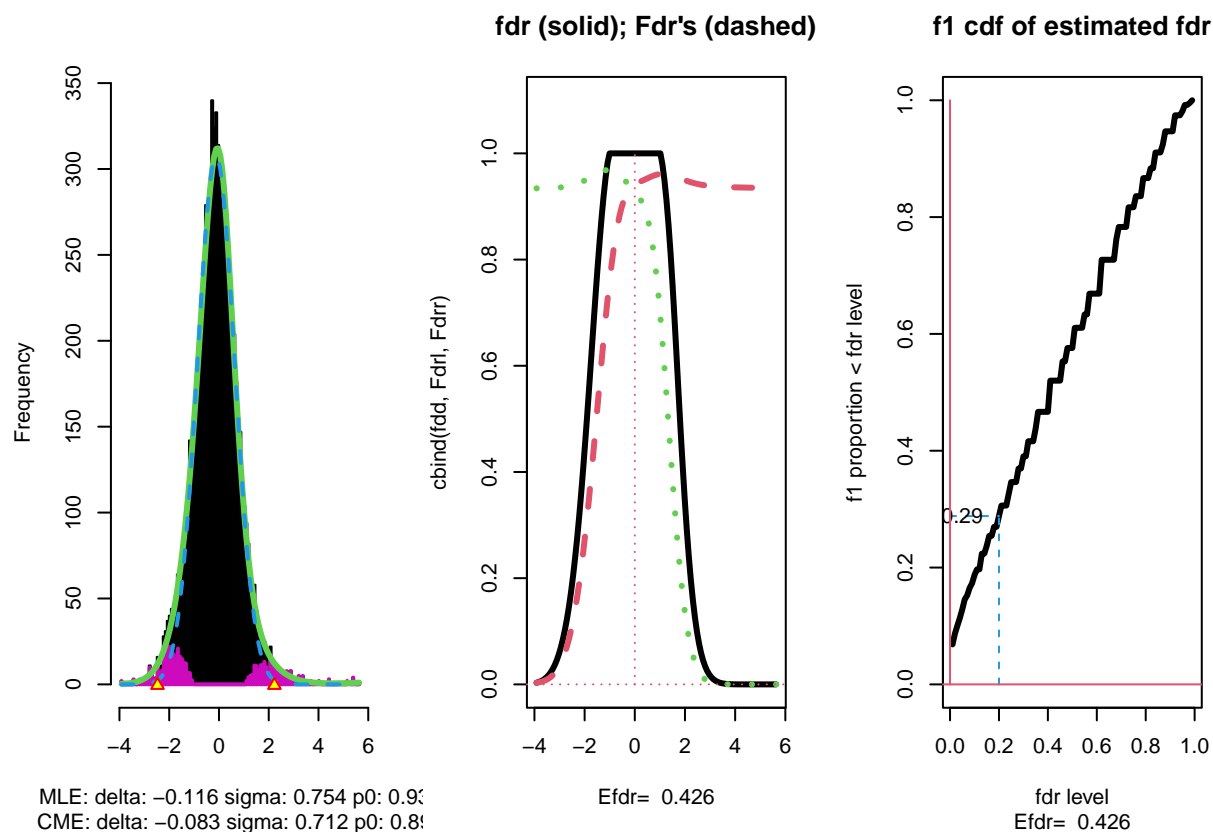
where Z is the test statistic distributed under the alternative hypothesis (f_1).

- This probability $P_{f_1} [\text{fdr}(Z) < \alpha]$ is a kind of extension of the definition of the power of a test: it is the probability that a non-null can be detected when the nominal local fdr is set at α .
- The graph shows, for examples, that with $\alpha = 0.20$ we only have $P_{f_1} [\text{fdr}(Z) < \alpha] = 0.24$, i.e. only 24% of the non-nulls are expected to be discovered.
- At the bottom of the graph we read $\text{Efdr} = 0.486$. Hence, the local fdr for a typical non-null feature is expected to be 48.6% which is rather large. The study is not well powered!

3.6 Comparison with gene expression study

- HIV dataset: 7680 z-values, each relating to a two-sample t-test comparing gene expression of 4 normal to 4 HIV patients.

```
data(hivdata)
res <- locfdr(hivdata, plot=4)
```



```
mean(res$fdr[res$fdr<0.2])
```

```
#> [1] 0.07539095
```

Session info

Session info

```
#> [1] "2024-10-03 06:24:55 UTC"
```

```
#> - Session info -----
#> setting  value
#> version  R version 4.1.3 (2022-03-10)
#> os       Ubuntu 22.04.5 LTS
#> system    x86_64, linux-gnu
#> ui       X11
#> language (EN)
#> collate   C.UTF-8
#> ctype     C.UTF-8
#> tz        UTC
#> date      2024-10-03
#> pandoc    3.1.11 @ /opt/hostedtoolcache/pandoc/3.1.11/x64/ (via rmarkdown)
#>
```

```

#> - Packages -----
#> ! package      * version date (UTC) lib source
#> P assertthat    0.2.1   2019-03-21 [?] CRAN (R 4.1.3)
#> P backports     1.4.1   2021-12-13 [?] CRAN (R 4.1.3)
#> P BiocManager   1.30.16  2021-06-15 [?] CRAN (R 4.1.3)
#> P bit           4.0.4   2020-08-04 [?] CRAN (R 4.1.3)
#> P bit64         4.0.5   2020-08-30 [?] CRAN (R 4.1.3)
#> P bookdown      0.24    2021-09-02 [?] CRAN (R 4.1.3)
#> P broom         0.7.11  2022-01-03 [?] CRAN (R 4.1.3)
#> P bslib         0.3.1   2021-10-06 [?] CRAN (R 4.1.3)
#> P cellranger    1.1.0   2016-07-27 [?] CRAN (R 4.1.3)
#> P cli           3.1.1   2022-01-20 [?] CRAN (R 4.1.3)
#> P colorspace    2.0-2   2021-06-24 [?] CRAN (R 4.1.3)
#> P crayon        1.4.2   2021-10-29 [?] CRAN (R 4.1.3)
#> P curl          4.3.2   2021-06-23 [?] CRAN (R 4.1.3)
#> P DBI           1.1.2   2021-12-20 [?] CRAN (R 4.1.3)
#> P dbplyr        2.1.1   2021-04-06 [?] CRAN (R 4.1.3)
#> P digest        0.6.29  2021-12-01 [?] CRAN (R 4.1.3)
#> P dplyr         * 1.0.7   2021-06-18 [?] CRAN (R 4.1.3)
#> P ellipsis      0.3.2   2021-04-29 [?] CRAN (R 4.1.3)
#> P evaluate      0.14    2019-05-28 [?] CRAN (R 4.1.3)
#> P fansi         1.0.2   2022-01-14 [?] CRAN (R 4.1.3)
#> P farver        2.1.0   2021-02-28 [?] CRAN (R 4.1.3)
#> P fastmap       1.1.0   2021-01-25 [?] CRAN (R 4.1.3)
#> P forcats       * 0.5.1   2021-01-27 [?] CRAN (R 4.1.3)
#> P fs            1.5.2   2021-12-08 [?] CRAN (R 4.1.3)
#> P generics      0.1.1   2021-10-25 [?] CRAN (R 4.1.3)
#> P gganimate     * 1.0.7   2020-10-15 [?] CRAN (R 4.1.3)
#> P ggplot2       * 3.3.5   2021-06-25 [?] CRAN (R 4.1.3)
#> P glue          1.6.1   2022-01-22 [?] CRAN (R 4.1.3)
#> P gridExtra     * 2.3     2017-09-09 [?] CRAN (R 4.1.3)
#> P gtable        0.3.0   2019-03-25 [?] CRAN (R 4.1.3)
#> P haven         2.4.3   2021-08-04 [?] CRAN (R 4.1.3)
#> P highr         0.9     2021-04-16 [?] CRAN (R 4.1.3)
#> P hms           1.1.1   2021-09-26 [?] CRAN (R 4.1.3)
#> P htmltools     0.5.2   2021-08-25 [?] CRAN (R 4.1.3)
#> P httr          1.4.2   2020-07-20 [?] CRAN (R 4.1.3)
#> P jquerylib     0.1.4   2021-04-26 [?] CRAN (R 4.1.3)
#> P jsonlite      1.7.3   2022-01-17 [?] CRAN (R 4.1.3)
#> P knitr         1.37    2021-12-16 [?] CRAN (R 4.1.3)
#> P labeling      0.4.2   2020-10-20 [?] CRAN (R 4.1.3)
#> P lifecycle     1.0.1   2021-09-24 [?] CRAN (R 4.1.3)
#> P locfdr        * 1.1-8   2015-07-15 [?] CRAN (R 4.1.3)
#> P lubridate     1.8.0   2021-10-07 [?] CRAN (R 4.1.3)
#> P magick        * 2.7.3   2021-08-18 [?] CRAN (R 4.1.3)
#> P magrittr      2.0.2   2022-01-26 [?] CRAN (R 4.1.3)
#> P modelr        0.1.8   2020-05-19 [?] CRAN (R 4.1.3)
#> P munsell       0.5.0   2018-06-12 [?] CRAN (R 4.1.3)
#> P pillar        1.6.5   2022-01-25 [?] CRAN (R 4.1.3)
#> P pkgconfig     2.0.3   2024-10-02 [?] Github (r-lib/pkgconfig@b81ae03)
#> P plyr          1.8.6   2020-03-03 [?] CRAN (R 4.1.3)
#> P prettyunits   1.1.1   2020-01-24 [?] CRAN (R 4.1.3)
#> P progress      1.2.2   2019-05-16 [?] CRAN (R 4.1.3)
#> P purrr         * 0.3.4   2020-04-17 [?] CRAN (R 4.1.3)

```



```

#> P R6                2.5.1    2021-08-19 [?] CRAN (R 4.1.3)
#> P Rcpp               1.0.8    2022-01-13 [?] CRAN (R 4.1.3)
#> P readr              * 2.1.1    2021-11-30 [?] CRAN (R 4.1.3)
#> P readxl             1.3.1    2019-03-13 [?] CRAN (R 4.1.3)
#>   renv                0.15.2    2022-01-24 [1] CRAN (R 4.1.3)
#> P reprex             2.0.1    2021-08-05 [?] CRAN (R 4.1.3)
#> P rlang               1.0.0    2022-01-26 [?] CRAN (R 4.1.3)
#> P rmarkdown          2.11     2021-09-14 [?] CRAN (R 4.1.3)
#> P rstudioapi         0.13     2020-11-12 [?] CRAN (R 4.1.3)
#> P rvest               1.0.2    2021-10-16 [?] CRAN (R 4.1.3)
#> P sass                0.4.0    2021-05-12 [?] CRAN (R 4.1.3)
#> P scales              1.1.1    2020-05-11 [?] CRAN (R 4.1.3)
#> P sessioninfo        1.2.2    2021-12-06 [?] CRAN (R 4.1.3)
#> P stringi            1.7.6    2021-11-29 [?] CRAN (R 4.1.3)
#> P stringr            * 1.4.0    2019-02-10 [?] CRAN (R 4.1.3)
#> P tibble              * 3.1.6    2021-11-07 [?] CRAN (R 4.1.3)
#> P tidyr               * 1.1.4    2021-09-27 [?] CRAN (R 4.1.3)
#> P tidyselect          1.1.1    2021-04-30 [?] CRAN (R 4.1.3)
#> P tidyverse          * 1.3.1    2021-04-15 [?] CRAN (R 4.1.3)
#> P tweenr              1.0.2    2021-03-23 [?] CRAN (R 4.1.3)
#> P tzdb                0.2.0    2021-10-27 [?] CRAN (R 4.1.3)
#> P utf8                1.2.2    2021-07-24 [?] CRAN (R 4.1.3)
#> P vctrs               0.3.8    2021-04-29 [?] CRAN (R 4.1.3)
#> P vroom               1.5.7    2021-11-30 [?] CRAN (R 4.1.3)
#> P withr               2.4.3    2021-11-30 [?] CRAN (R 4.1.3)
#> P xfun                0.29     2021-12-14 [?] CRAN (R 4.1.3)
#> P xml2                1.3.3    2021-11-30 [?] CRAN (R 4.1.3)
#> P yaml                2.2.2    2022-01-25 [?] CRAN (R 4.1.3)
#>
#> [1] /home/runner/work/HDDA23/HDDA23/renv/library/R-4.1/x86_64-pc-linux-gnu
#> [2] /opt/R/4.1.3/lib/R/library
#>
#> P -- Loaded and on-disk path mismatch.
#>
#> -----

```