More concerns about the z-curve method

Erik van Zwet

14 januari, 2026

Contents

This report accompanies a blog post More concerns about the z-curve method.

1 Packages

```
suppressPackageStartupMessages({
  library(ggplot2)
  library(zcurve)
})
```

2 Introduction

Suppose we have an unbiased, normally distributed estimator y with standard error s of some (unknown) parameter of interest θ. We assume that s is known. The difference between knowing s and having to estimate it, is the difference between a t-test and a z-test. That's a minor difference when there aren't any serious outliers and the sample size is not too small. So, we assume that y has the normal distribution with mean θ and standard deviation s. The 95% confidence interval for θ is $y \pm 1.96 \cdot s$. All very standard.

The z-statistic is the ratio of the estimator to its standard error, so $z = y/s$. It follows that z has the normal distribution with mean θ/s and standard deviation 1. We refer to θ/s, i.e. the ratio of the true parameter to the standard error of its estimator, as the signal-to-noise ratio or SNR. If we define $\text{SNR} = \theta/s$. Then the z-statistic has the normal distribution with mean SNR and standard deviation 1.

If we have a collection of studies, then the distribution of the z-statistics is the convolution (sum) of the distribution of the SNRs of the studies and the standard normal distribution. If we've estimated the distribution of the z-statistics, we can get the distribution of the SNRs by *deconvolution*. Deconvolution is known to be very unstable. That means that we would need very many data points (studies) or very strong assumptions -– preferably both — to get an accurate result.

The z-curve method (Brunner and Schimmack (2020), Bartos and Schimmack (2022), Schimmack and Bartos (2023)) is based on the assumption that the absolute values of the SNRs have a discrete distribution supported on $0, 1, 2, \dots, 6$. We denote the associated probabilities by $p_0, p_1, \dots, p_6$. Note that SNR=0 corresponds to a "null effect" so p_0 is the proportion of null effects among the studies. The distribution of the absolute z-statistics is a mixture of the folded versions of normal distributions with means $0, 1, 2, \dots, 6$ and unit variances.

To circumvent the effects of selection on statistical significance, z-curve uses only the absolute values of the z-statistics which exceed 1.96 in magnitude to estimate the p_i. The authors also truncate absolute z-statistics at 6. They then re-parameterize in terms of a mixture of folded normals truncated to $[1.96, 6]$. The new mixture weights $w_0, w_1, \dots, w_6$ are

$$w_i = \frac{p_i \, P(1.96 < |z| < 6 \mid \text{SNR} = i)}{\sum_{j=0}^{6} p_j P(1.96 < |z| < 6 \mid \text{SNR} = j)}.$$

Clearly, there is a 1-1 relation between the p_i and w_i since

$$p_i = \frac{w_i / P(1.96 < |z| < 6 \mid \text{SNR} = i)}{\sum_{j=0}^{6} w_j / P(1.96 < |z| < 6 \mid \text{SNR} = j)}.$$

Therefore maximum likelihood estimation of the w_i is equivalent to maximum likelihood estimation of the p_i.

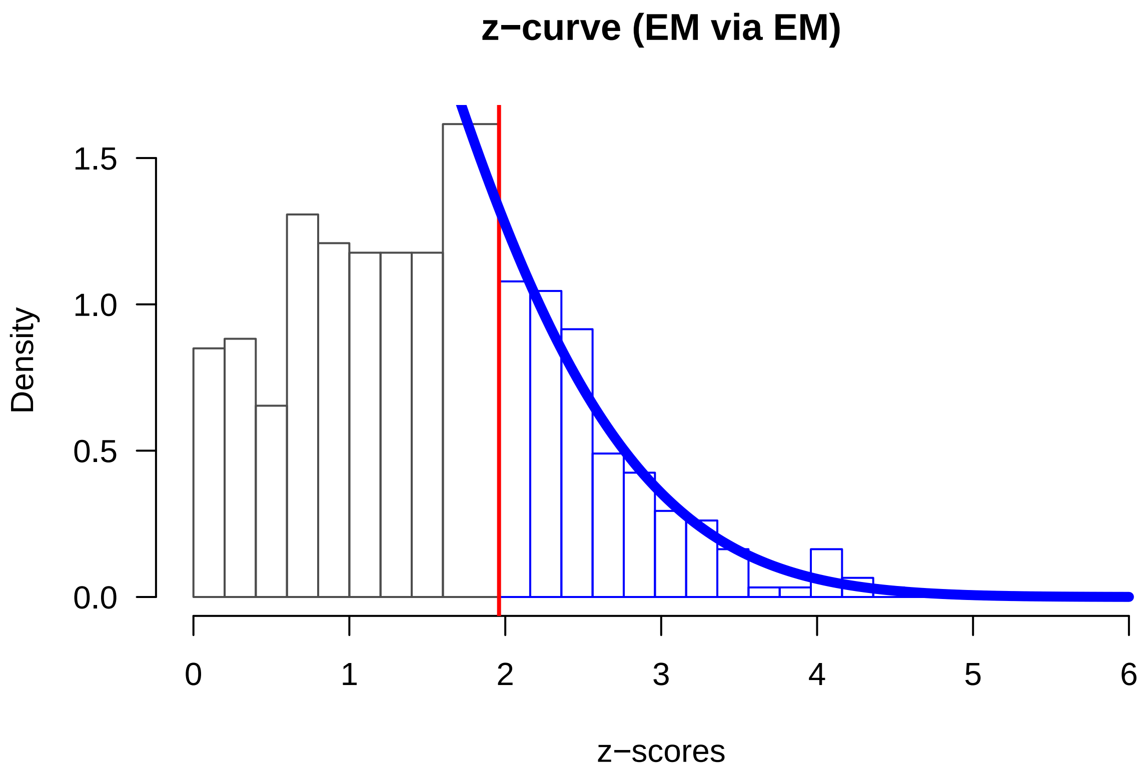
3 An example

The function `rmix()` samples from normal mixture distributions with weights `p`, means `m` and standard deviations `s`.

```
rmix = function(n,p,m,s){
  d=rmultinom(n,1,p)
  rnorm(n,m%*%d,s%*%d)
}
```

Suppose we have a sample of size $n = 500$ from the normal distribution with mean 1.5 and standard deviation 1. The following code is more general because it also applies to normal mixtures.

```
set.seed(111)
# sample n=500 from mixture N(1.5,1)
n=500
p=c(0,1,0,0,0,0,0)
m=0:6;  m[2]=1.5
s=rep(1,7)
z=rmix(n,p,m,s)
fit=zcurve(z)
plot(fit)
```



The figure is typical; the zero mean component is essentially invisible from the observations exceeding 1.96.

4 Coverage of the mixture weights

The model is parameterized in terms of the mixture weights w_i of the truncated folded normals. These weights are

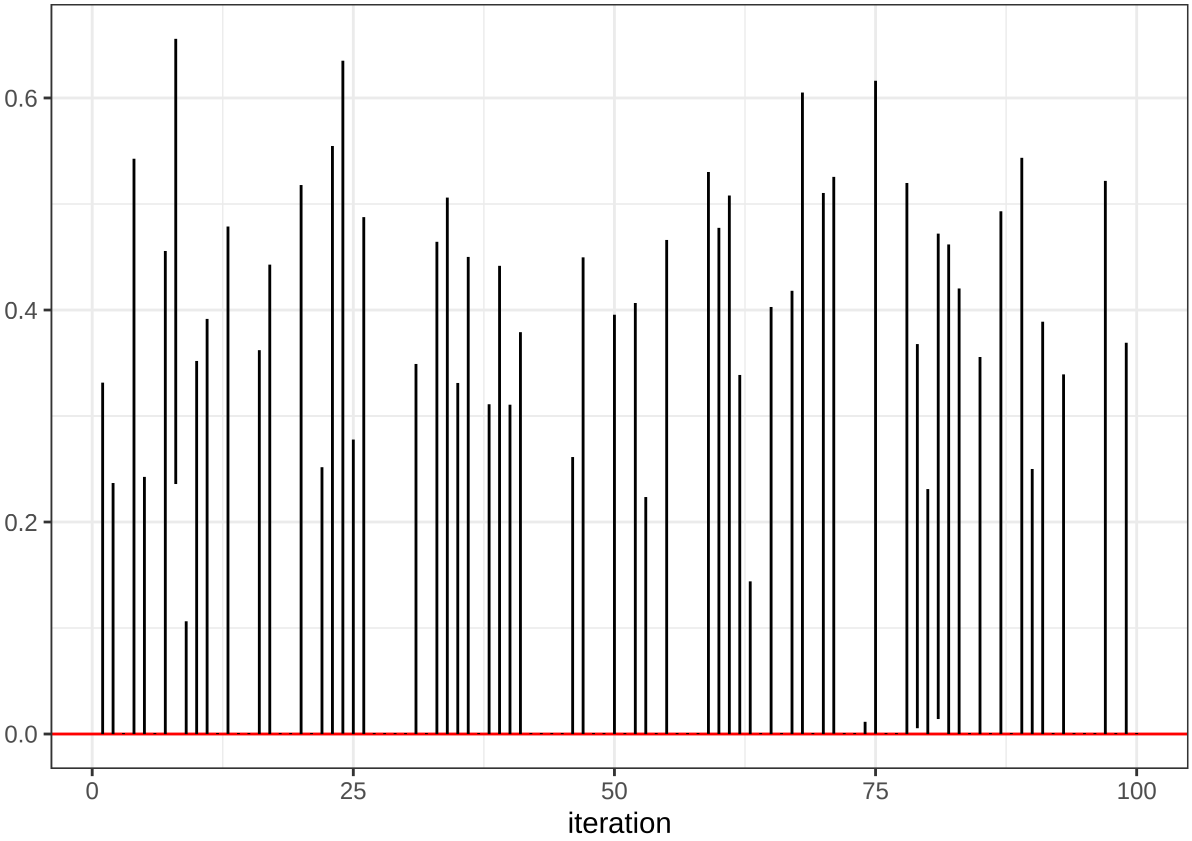
```
# prob = P(1.96 < |z| < 6 | SNR=m)
prob=pnorm(6, m, 1) - pnorm(1.96, m, 1) + pnorm(-1.96, m, 1) - pnorm(-6, m, 1)
w=p*prob; w=w/sum(w)
round(w,3)
```

```
## [1] 0 1 0 0 0 0 0
```

The function `zcurve` provides a 95% confidence interval for the mixture weight of the zero-mean component which in this example is equal to 0. We check the coverage of these intervals across 100 simulations.

```
k=100
ci_w1=ci_EDR=ci_ERR=ci_FDR=NULL
for (i in 1:k){
  z=rmix(n,p,m,s)
  fit=zcurve(z)
  ci_w1=rbind(ci_w1,summary(fit,type="parameters")$coefficients[1,]) # CI for w[1]
  ci_ERR=rbind(ci_ERR,summary(fit, all=TRUE)$coefficients[1,])       # CI for ERR
  ci_EDR=rbind(ci_EDR,summary(fit, all=TRUE)$coefficients[2,])       # CI for EDR
}
ci=data.frame(iteration=1:k,lower=ci_w1[,3],upper=ci_w1[,4])
mean((ci$lower < w[1]+0.0001) & (ci$upper > w[1]) )               # coverage of w[1]
```

```
## [1] 0.82
```



5 Coverage of the EDR

The EDR is the “population mean true power”. It “equals the overall probability of a significant result” (Brunner and Schimmack, 2020).

$$\mathrm{EDR} = P(|z| > 1.96) = \sum_{i=0}^{6} p_i \, P(|z| > 1.96 \mid \mathrm{SNR} = i).$$

```
pmix = function(x,p,m,s){     # cdf of normal mixture (vector x)
  drop(p %*% sapply(x, function(x) pnorm(x,mean=m,sd=s)))
}
EDR=pmix(-1.96,p,m,s) + 1 - pmix(1.96,p,m,s)
EDR
```

```
## [1] 0.3230282
```

We find the EDR is 0.323. The coverage across 100 iterations is

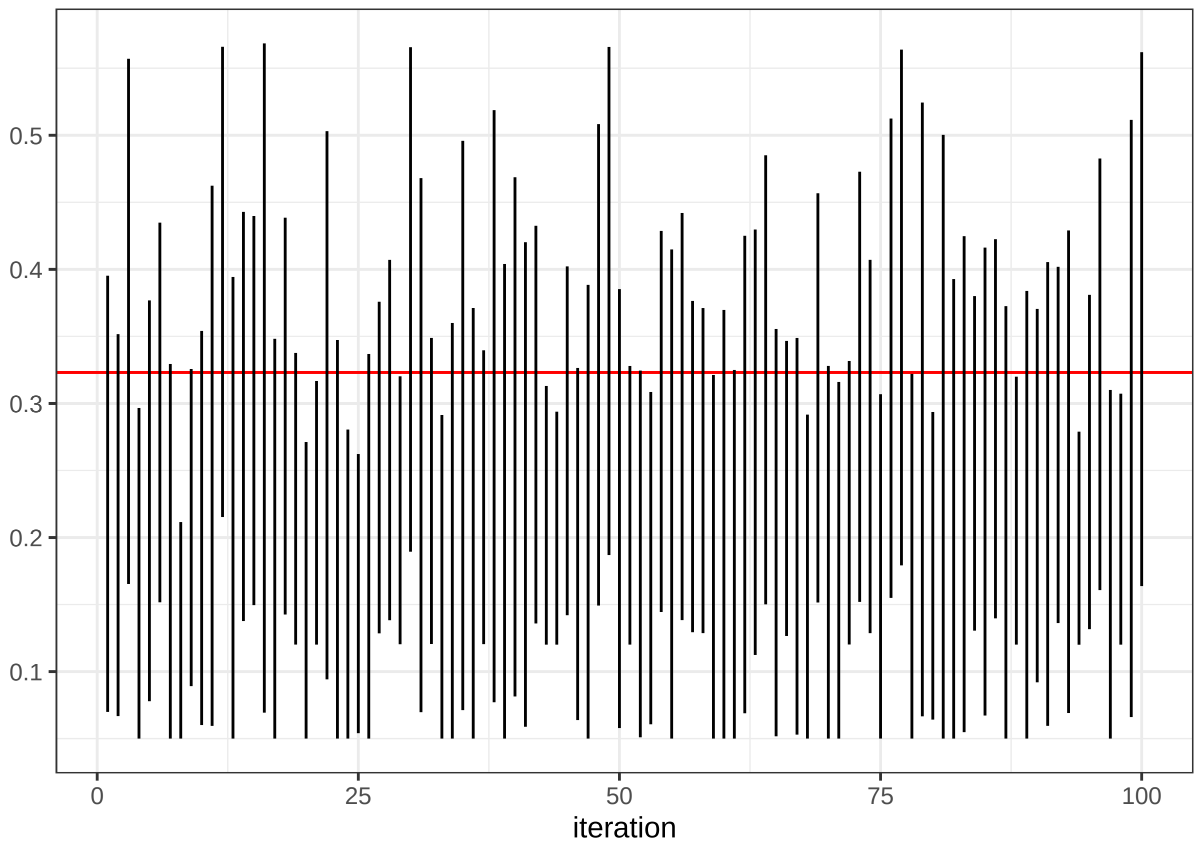
```
ci=data.frame(iteration=1:k,lower=ci_EDR[,2],upper=ci_EDR[,3])
mean((ci$lower < EDR) & (ci$upper > EDR) )     # coverage of EDR
```

```
## [1] 0.79
```

There is substantial bias:

```
estimate=ci_EDR[,1]
t.test(estimate,mu=EDR)
```

```
##
##  One Sample t-test
##
## data:  estimate
## t = -18.95, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0.3230282
## 95 percent confidence interval:
##  0.2114126 0.2325713
## sample estimates:
## mean of x
## 0.2219919
```


0.5
0.4
0.3
0.2
0.1
0
25
50
75
100
iteration

6 Coverage of the ERR

The ERR is the expected replication rate. It's the conditional probability of statistical significance of a replication study given statistical significance of the original study. If z_{repl} is an independent copy of z then

$$\mathrm{ERR} = P(|z_{\mathrm{repl}}| > 1.96 \text{ and } z_{\mathrm{repl}} \times z > 0 \mid |z| > 1.96)$$

We comute the ERR by Monte Carlo integration.

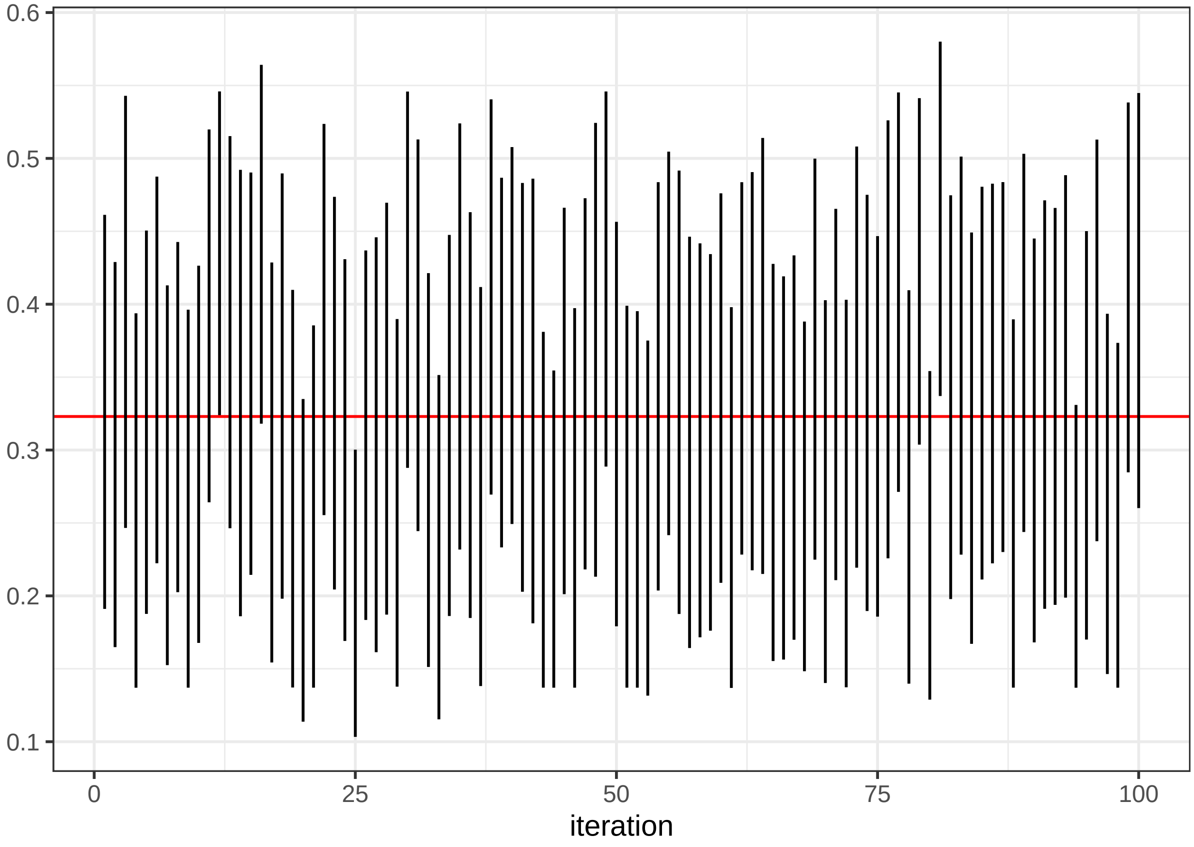
```
SNR=sample(m,10^6,replace=TRUE,prob=p)
z1=SNR+rnorm(10^6)
z2=SNR+rnorm(10^6)
signif=which(abs(z1)>1.96)
ERR=mean(abs(z2[signif])>1.96 & z1[signif]*z2[signif]>0)
ERR
```

```
## [1] 0.3229976
```

We find the ERR is 0.323. The coverage across 100 iterations is

```
ci=data.frame(iteration=1:k,lower=ci_ERR[,2],upper=ci_ERR[,3])
mean((ci$lower < ERR) & (ci$upper > ERR) )     # coverage of EDR
```

```
## [1] 0.97
```


iteration