A large-scale in silico replication project of ecological and evolutionary studies

Yefeng Yang, Erik van Zwet, Nikolaos Ignatiadis, Shinichi Nakagawa

Table of Contents

# Packages

knitr::opts\_chunk$set(fig.height = 8, fig.width = 10)  
set.seed(2023)  
suppressPackageStartupMessages({  
 library(dplyr)  
 library(readr)  
 library(tidyr)   
 library(ggplot2)  
 library(aplot)  
 library(ggplotify)   
 library(here)  
 library(patchwork)  
 library(plot3D)  
 })  
# functions  
source(here("func","func.R"))

# Data

The database comprised 466 meta-analytic datasets and 111,321 observations of ecological and evolutionary effects, curated by Costello and Fox. These datasets were obtained through a systematic search of meta-analysis papers indexed in Web of Science categories relevant to ecology and evolutionary biology. We eliminated effect size estimates with zero and missing sampling variances.

The 466 meta-analytic datasets encompassed diverse research topics within ecology and evolutionary biology. However, it is important to realize that the trials in our dataset are not a random sample of all the trials in the research field, and may therefore not be entirely representative.

Furthermore, it is well known that trials that do not reach statistical significance have a lower chance of publication. We do not know the extent of this “publication bias” and we do not take it into account.

Import the main data (the .csv file named main\_dat.csv).

dat\_all <- read.csv(here("data/main","main\_dat.csv"))   
head(dat\_all)

## paper.id meta.analysis.paper meta.analysis.id meta.analysis.year  
## 1 1 bird.et.al.2019.ecoletts 1 2019  
## 2 1 bird.et.al.2019.ecoletts 1 2019  
## 3 1 bird.et.al.2019.ecoletts 1 2019  
## 4 1 bird.et.al.2019.ecoletts 1 2019  
## 5 1 bird.et.al.2019.ecoletts 1 2019  
## 6 1 bird.et.al.2019.ecoletts 1 2019  
## response.variable study study.id study.id.cumulative study.year  
## 1 various Harrison\_1 1 1 1964  
## 2 various Harrison\_1 1 1 1964  
## 3 various Harrison\_1 1 1 1964  
## 4 various Harrison\_1 1 1 1964  
## 5 various Harrison\_1 1 1 1964  
## 6 various Harrison\_1 1 1 1964  
## id.effect.within.study eff.size.measure eff.size var.eff.size  
## 1 1 hedges.d -1.7771359 0.1099483  
## 2 2 hedges.d -1.7519846 0.2385344  
## 3 3 hedges.d -1.3276771 1.7153664  
## 4 4 hedges.d -1.3276771 1.7153664  
## 5 5 hedges.d -0.3359459 0.4703529  
## 6 6 hedges.d -0.6486920 1.0548881  
## weighted.eff.size weight  
## 1 -16.1633810 9.0951856  
## 2 -7.3447870 4.1922670  
## 3 -0.7739904 0.5829658  
## 4 -0.7739904 0.5829658  
## 5 -0.7142422 2.1260631  
## 6 -0.6149392 0.9479678

Clean the dataset

# basic cleaning   
dat\_all <- dat\_all[!is.na(dat\_all$eff.size) & !is.na(dat\_all$var.eff.size), ] # remove NAs  
dat\_all <- dat\_all[dat\_all$var.eff.size != 0, ] # remove negative and zero variance (sampling variance should be positive)  
  
# classify effect size measures  
grouped\_es <- NA  
grouped\_es[dat\_all$eff.size.measure == "cohens.d"|  
 dat\_all$eff.size.measure == "hedges.d"|  
 dat\_all$eff.size.measure == "hedges.g"|  
 dat\_all$eff.size.measure == "abs.hedges.d"|  
 dat\_all$eff.size.measure == "SMD"] <- c("SMD") # (1) SMD: cohens.d, hedges.d, hedges.g, abs.hedges.d, SMD  
grouped\_es[dat\_all$eff.size.measure == "log.ratio"] <- c("lnRR") # (2) lnRR: log.ratio  
grouped\_es[dat\_all$eff.size.measure == "z.r"] <- c("Zr") # (3) Zr: z.r  
# (4) binary: IRR, log.odds.ratio  
#grouped\_es[dat\_all$eff.size.measure == "IRR" |   
# dat\_all$eff.size.measure == "log.odds.ratio"] <- c("binary")   
grouped\_es[dat\_all$eff.size.measure == "mean.diff" |  
 dat\_all$eff.size.measure == "reg.slope" |   
 dat\_all$eff.size.measure == "IRR" |   
 dat\_all$eff.size.measure == "log.odds.ratio"] <- c("uncommon") # (5) uncommon: mean.diff, reg.slope  
dat\_all$grouped\_es <- as.factor(grouped\_es)  
  
  
# only select relevant variables  
dat <- dat\_all %>% dplyr::select(paper.id, meta.analysis.paper, meta.analysis.id, meta.analysis.year, study, study.year, eff.size.measure, grouped\_es, eff.size, var.eff.size)  
  
dat <- dat %>% mutate(se.eff.size = sqrt(var.eff.size)) # create the variable of sampling error  
dat <- dat %>% mutate(z = eff.size/se.eff.size) # calculate z value  
  
# create a variable to label each MA  
dat$MA\_case <- paste("MA", dat$meta.analysis.id, sep = "\_")  
  
# make a copy  
d <- dat  
  
nrow(d) # number of estimates

## [1] 111321

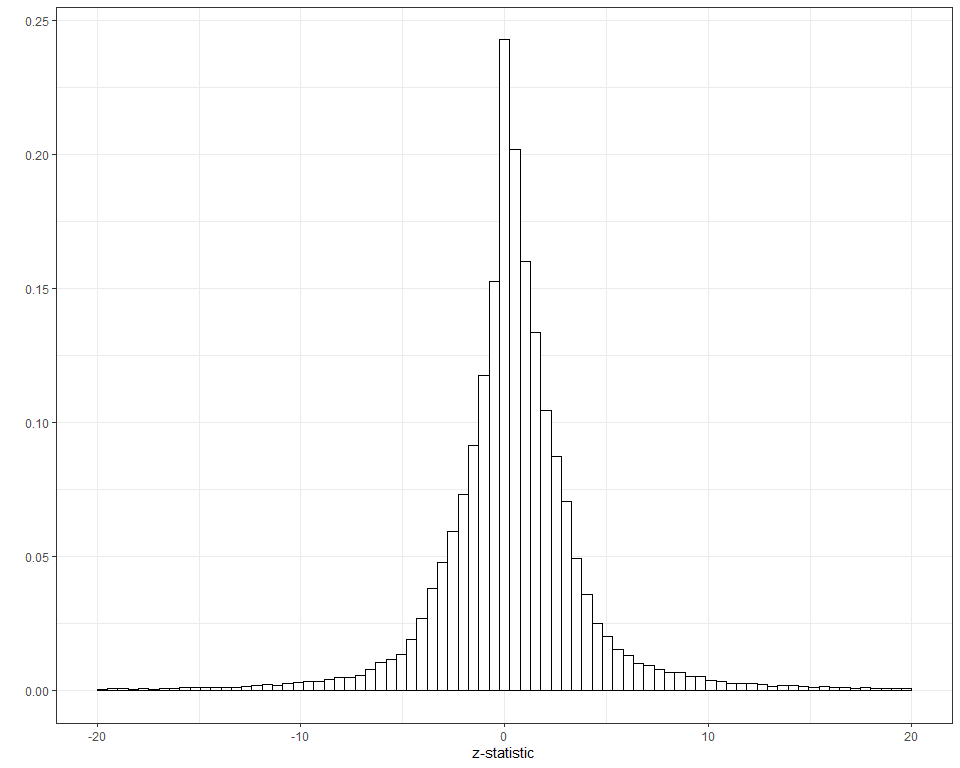
length(unique(dat$study)) # number of unique studies

## [1] 14630

length(unique(dat$meta.analysis.id)) # number of meta-analyses

## [1] 466

We plot the -statistics.



# Modelling

For each observation in the database, let denote the (unobservable!) true or population effect size, and be the observed effect size. We assume that a normally distributed, unbiased estimate of the true effect with a known sampling standard deviation (aka standard error) . So,

$$\overline{ES} \sim {\cal N}(ES,SE^2)$$

The main effect size measures in our database are standardized mean differences (SMD; 45%), log-transformed response ratios (lnRR; 36%), and Fisher’s z-transformed correlations (15%). For these measures, it is reasonable (and customary) to assume normality. The z-statistic is defined as

If the absolute value of the z-statistic exceed 1.96 then the observed effect is “statistically significantly” different from zero at significance level (two-sided).

Finally, we define the signal-noise-ratio () as the strength of true effect size (the signal) relative standard error of the estimate (the noise),

It follows from our assumptions that

$$z \sim {\cal N}(SNR,1)$$

In other words, we can think if the z-statistic as the sum of the signal-to-noise ratio and a standard normal error term.

## z-statistic

We start by estimating the distribution of the z-statistics across the trials in our dataset. We use the method of maximum likelihood. We will assume that the z-statistics follow a mixture of zero-mean normal distributions. We can write this distribution as

where is the density of the standard normal distribution, and the mixture probabilities (or “weights”) are non-negative and sum to one. That is, $p\_i $ and .

By fitting a symmetric distribution to the observed -statistics, we are ignoring their signs. Put differently, by fitting a mixture of zero-mean normals to the z-stats, we are essentially fitting a mixture of half-normals to the *absolute* z-stats.

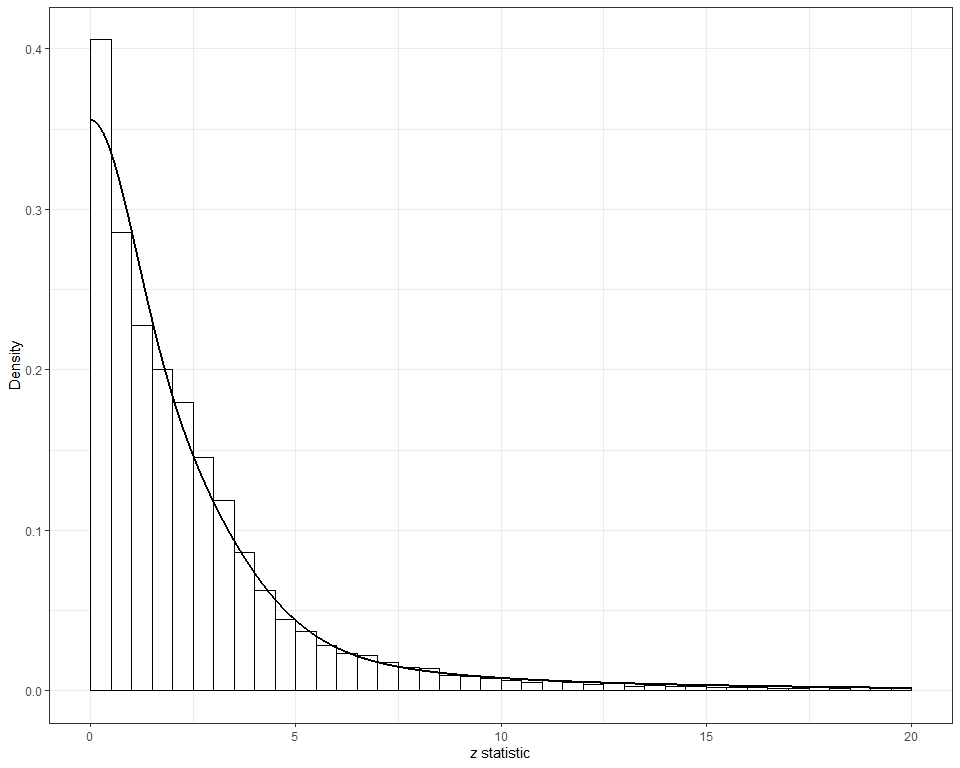
Since the z-statistics are the sum of the signal-to-noise ratio and a standard normal error term, the standard deviations of the mixture components must be at least 1. So, we run a constrained optimization to find the maximum likelihood estimates. The constraints are:

Some studies have multiple outcomes, so we weight each study inversely proportional to how often it appears in our data set.

# count number of outcomes per study  
d = group\_by(d,study) %>% mutate(count=n(),w=1/count)   
# censor two extreme z values at, say, -100 and 100, respectively  
d$z[d$z == max(d$z)] = 100  
d$z[d$z == min(d$z)] = -100  
  
df=mix(z=d$z,k=4,c1=0,c2=10,weights=d$w)  
p=df$p  
m=df$m  
sigma=df$sigma  
round(df,2)

## p m sigma  
## 1 0.57 0 2.40  
## 2 0.16 0 1.00  
## 3 0.15 0 4.50  
## 4 0.11 0 10.87

We plot the *weighted* histogram of the absolute -statistics together with the fitted mixture. We see that the mixture fits quite well.



## SNR

As we discussed, the z-statistics are the sum of the signal-to-noise ratio and a standard normal error term. Therefore it is easy to obtain the distribution of the SNRs from the distribution of the -statistics. We simply subtract 1 from variances of the mixture components. So, the distribution of the SNRs can be written as

where .

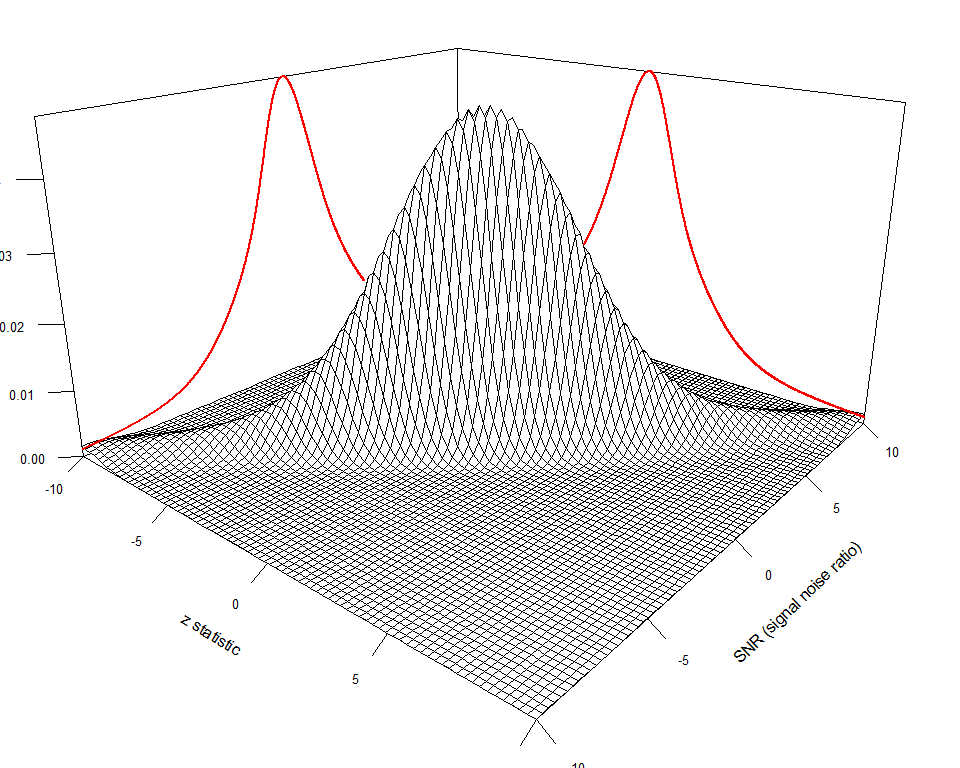
## Joint distribution

We have now estimated the marginal distributions of the z-statistics and the SNRs. But we also know the *conditional* distribution of the z-statistic given the SNR (it’s normal with mean SNR and standard deviation 1). This means that we have the *joint* distribution of the z-statistics the SNRs.

Finally, we can use the well-known theory of bivariate normal distributions to obtain the conditional distribution of the SNR given the observed z-statistic.

We show the joint distribution of z and SNR.

djoint <- function(x,y,p,m,s) {   
 dmix(y,p,m,s)\*dnorm(x-y,mean=0,sd=1) # conditional on SNR, z ~ N(SNR,1)  
}  
  
x <- seq(-10, 10, length.out = 80)  
y <- x   
z <- outer(x, y, djoint, p=p,m=m,s=sqrt(sigma^2-1))  
# set margins  
par(oma = c(0, 0, 1, 1)) # set the outer margin to control the overall space around the plot  
par(mar = c(1, 0, 0, 0)) # set the inner margin (mar) - especially the bottom  
# joint distribution  
zlim <- c(0, 0.048) # set the z-axis limits  
# define custom z-tick positions  
zticks <- seq(zlim[1], zlim[2], length.out = 5)  
# trans3d is used to convert the 3D coordinates to 2D screen coordinates, and text is used to add text annotations.   
joint.3D <- persp3D(x = x,   
 y = y,  
 z = z,  
 theta = 40, phi = 20, expand = 0.6,   
 facets = FALSE,   
 col="black",  
 xlab = "z statistic", ylab = "SNR (signal noise ratio)",zlab="Density",  
 ticktype="detailed", nticks = 5,  
 cex.lab = 1,  
 cex.axis = 0.8,  
 zlim = zlim,  
 zticks = list(at = zticks))   
text(trans3d(x = min(x) + 2.5, y = min(y) - 3.8, z = 0.052, joint.3D), labels = "", font = 2, cex = 1.3)   
  
# marginal distribution of z (up to scale factor)  
x\_z=seq(-2.8,10,0.01)  
z\_z=drop(p %\*% sapply(x\_z, function(x) dnorm(x,mean=m,sd=sigma)))/3.7  
lines(trans3d(x\_z,y=10,z=z\_z,joint.3D),col="red",lwd=2)  
  
# marginal distribution of SNR (up to scale factor)  
y\_snz=seq(-10,3.8,0.01)  
z\_snz=drop(p %\*% sapply(y\_snz, function(x) dnorm(x,mean=m,sd=sigma)))/3.7   
lines(trans3d(x=-10,y=y\_snz,z=z\_snz,joint.3D),col="red",lwd=2)



# Estimates

We can use the estimated joint distribution of the z-statistics and the SNRs to calculate the statistical power and replicability for the 111,321 observed ecological and evolutionary effects.

## Power

The “achieved” or “actual” power is simply the probability of reaching statistical significance (two-sided, level ). Obviously, this probability depends on the true effect size and on the standard error of the estimate. Under our assumptions, this can be expressed in terms of the SNR.

where is the cumulative distribution function (CDF) of the standard normal distribution. Once again: This is the power against the *true* effect. Of course, the true effect is never known. However, since we can estimate the distribution of the SNR across the studies in our dataset, we can also estimate the distribution of the power across a large number of studies.

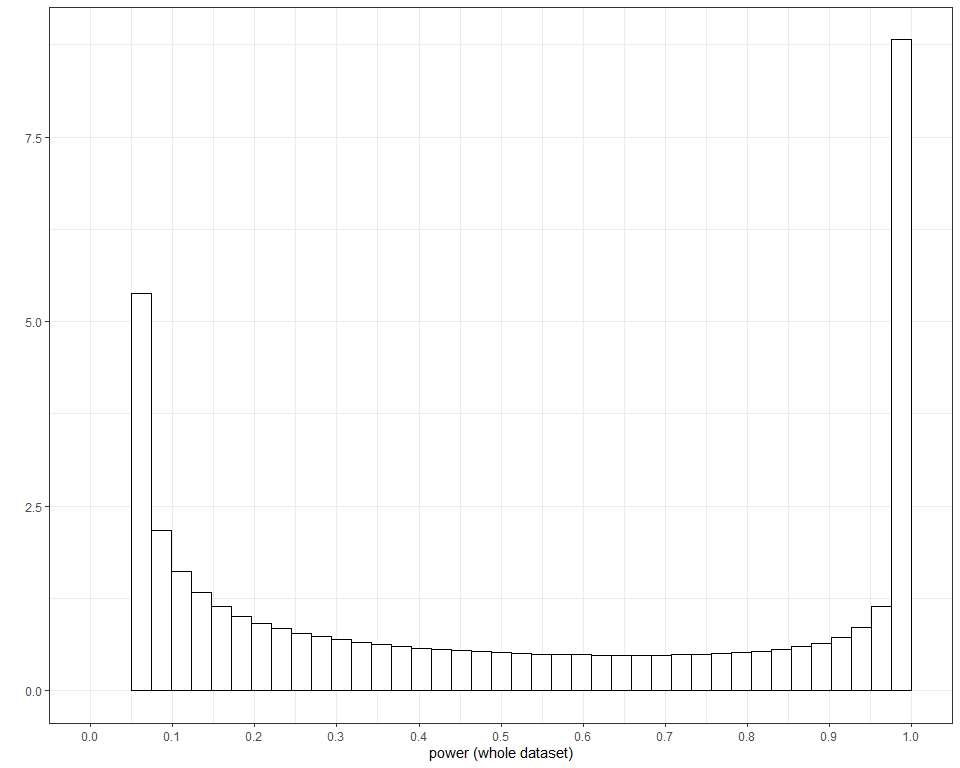
Note 1: The power against the true effect should not to be confused with the power the study is designed for. Most studies are designed to have 80% or 90% against some effect that is considered important or plausible (or both).

Note 2: The power against the true effect should also not be confused with the “observed” of “post hoc” power.

Since the power against the true effect is a function of the SNR, we can obtain its distribution by transforming the distribution of the SNR. We generate a sample of size from the (estimated) distribution of the SNRs, and then we apply the transformation to get a sample from the distribution of the power.

snr=rmix(10^7,p=p,m=m,s=sqrt(sigma^2 - 1))  
power=pnorm(-1.96,snr,1) + 1 - pnorm(1.96,snr,1)  
df=data.frame(power=power)

We can now show the power distribution with a histogram.



Based on the estimated distribution of the SNR, the average power can be calculated.

summary(power)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.05000 0.06427 0.31104 0.44294 0.87322 1.00000

We also have a direct estimate of the mean power, namely the weighted proportion of significant results.

sum(d$w\*(abs(d$z)>1.96))/sum(d$w)

## [1] 0.4565507

## Replication

Suppose we have conducted a study and obtained the z-statistic . Now consider a (hypothetical) replication study with exactly the same design (the same effect, sample size, standard error, etc.) Now consider the event that the replication is “successful” in the sense that it reaches statistical significance in the same direction as the original study, i.e.

Using simulation, we can easily compute mean replication for the whole data set, both unconditionally and conditionally on statistical significance

snr=rmix(10^7,p=p,m=m,s=sqrt(sigma^2 - 1))  
z.orig=snr + rnorm(10^7) # original  
z.repl = snr + rnorm(10^7) # replication  
replicate=(z.orig \* z.repl > 0) & (abs(z.repl) > 1.96)  
mean(replicate) # unconditional probability of replication

## [1] 0.4273329

mean(replicate[abs(z.orig)>1.96]) # conditional probability of replication given |z|>1.96

## [1] 0.7666828

We can also compute the conditional probability of “successful replication” given the absolute value of the -statistic of the original study. The symmetry of the distribution of the -statistic implies that the conditional probability of “successful replication” given is equal to the conditional probability given .

replicate=sapply(d$z,replcalc,p=p,m=m,s=sqrt(sigma^2-1))

We can verify the unconditional and conditional probability of “successful replication” by averaging over the empirical distribution of the observed -statistics.

sum(d$w\*replicate)/sum(d$w) # unconditional probability of replication

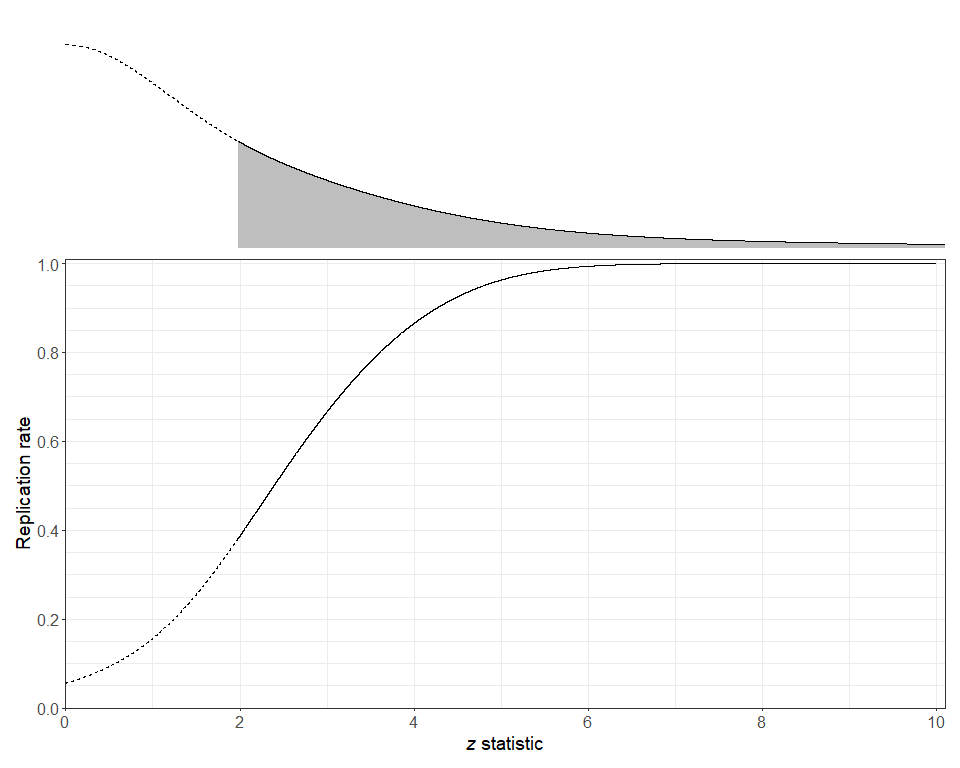
## [1] 0.4279977

ind=which(abs(d$z)>1.96)  
sum(d$w[ind]\*replicate[ind])/sum(d$w[ind]) # conditional probability of replication given |z|>1.96

## [1] 0.7574602

Replication rate curve.

# pdf of the z-stats  
density=dmix(d$z,p,m,sigma)  
  
df=data.frame(z=abs(d$z),replicate=replicate,density=density)  
  
replicate.dist <- ggplot(data=df, aes(x = z, y = replicate)) +   
 geom\_line(linetype="dashed") +  
 geom\_line(data=filter(df,z>1.96),linetype="solid") +   
 scale\_y\_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.2),  
 minor\_breaks=seq(0,1,0.05),   
 expand = expansion(mult = c(0, 0.01))) +  
 scale\_x\_continuous(limits = c(0, 10), breaks = seq(0, 10, by = 2),   
 expand = expansion(mult = c(0, 0.01))) +  
 ylab("Replication rate") + xlab(expression(paste(italic(z)~"statistic"))) +  
 labs(colour = NULL) +   
 theme\_bw() +   
 theme(axis.title = element\_text(size = 14),  
 axis.text = element\_text(size = 12))  
  
top.dist2 <- ggplot(data=df, aes(x = z,y=density)) +  
 geom\_line(linetype="dashed")+  
 geom\_ribbon(data=filter(df,z>1.96),aes(z, ymax=density, ymin=0),   
 fill="grey") +  
 geom\_line(data=filter(df,z>1.96),linetype="solid") +  
 scale\_x\_continuous(limits = c(0, 10), expand = expansion(mult = c(0, 0.01))) +  
 labs(x = "", y = "") +   
 theme\_void() +  
 theme(plot.margin = grid::unit(c(0, 0, 0, 0), "lines"))  
  
replicate.top.dist <- suppressWarnings(as.ggplot(insert\_top(  
 replicate.dist, top.dist2, height = 0.5)))   
replicate.top.dist + plot\_layout(tag\_level = 'new') +  
 plot\_annotation(tag\_levels = list(c(''))) &   
 theme(plot.tag = element\_text(size = 16, face = "bold"))



We show the conditional replication rates at selected z-values, re-formulated as the scales of evidence strength according to according to Bland M. 2015. An Introduction to Medical Statistics. Oxford, UK: Oxford Univ. Press. 4th ed. We covert z-values to two-sided p-values. Our purpose is to build up the intuitive relationship between the replication rates and evidence strength, albeit interpreting z-values (or, equivalently, p-values) as evidence strength is controversial (see **Notes**).

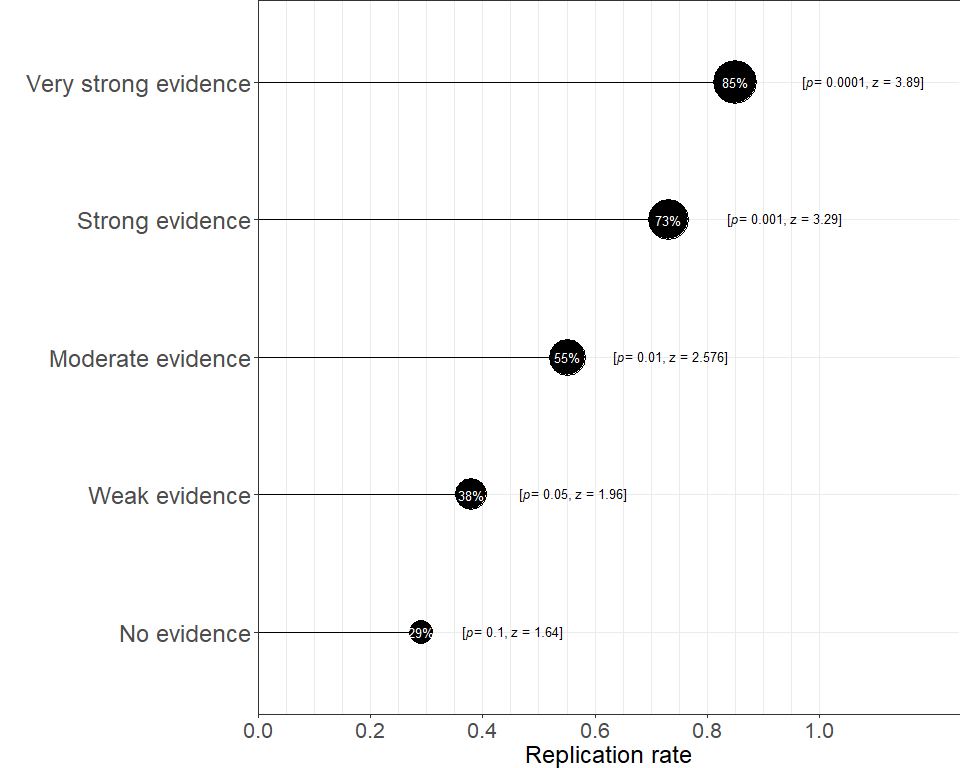
**Notes:**

The “scales of evidence” (like “no evidence”, “weak evidence”,…) are quite problematic, see Efron, B., Gous, A., Kass, R. E., Datta, G. S., & Lahiri, P. (2001). Scales of evidence for model selection: Fisher versus Jeffreys. Lecture Notes-Monograph Series, 208-256. It’s even doubtful if p-values can be taken as measures of evidence at all. As Wasserstein puts it: “By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis.” See Wasserstein, R. L., & Lazar, N. A. (2016). The ASA statement on p-values: context, process, and purpose. The American Statistician, 70(2), 129-133.

pval=c(0.1,0.05,0.01,0.001,0.0001)  
strength = c("No evidence", "Weak evidence","Moderate evidence",  
 "Strong evidence","Very strong evidence")  
strength = factor(strength,  
 levels=c("No evidence", "Weak evidence",  
 "Moderate evidence","Strong evidence",  
 "Very strong evidence"))  
zval=qnorm(1 - pval/2)  
replicate=sapply(zval,replcalc,p=p,m=m,s=sqrt(sigma^2-1))  
replicate=round(replicate,2)  
replicate\_typical <- data.frame(strength, replicate)  
data.frame(strength,pval,zval,replicate)

## strength pval zval replicate  
## 1 No evidence 1e-01 1.644854 0.29  
## 2 Weak evidence 5e-02 1.959964 0.38  
## 3 Moderate evidence 1e-02 2.575829 0.55  
## 4 Strong evidence 1e-03 3.290527 0.73  
## 5 Very strong evidence 1e-04 3.890592 0.85

# figure  
replicate\_typical.p <- ggplot(replicate\_typical, aes(x = replicate, y = strength)) +  
 geom\_segment(aes(x = 0, y = strength, xend = replicate, yend = strength)) +  
 geom\_point(aes(size = replicate)) +  
 scale\_size(range=c(8,15)) +  
 geom\_text(aes(label = scales::percent(replicate)), size = 3.5, color = "white") +   
 annotate("text", x = replicate[1] + 0.164, y = 1, label = 'paste("[", italic(p), "= 0.1,"~"z = 1.64", "]", sep = "")', parse = TRUE, size = 3.5) +  
annotate("text", x = replicate[2] + 0.181, y = 2, label = 'paste("[", italic(p), "= 0.05,"~"z = 1.96", "]", sep = "")', parse = TRUE, size = 3.5) +  
annotate("text", x = replicate[3] + 0.185, y = 3, label = 'paste("[", italic(p), "= 0.01,"~"z = 2.576", "]", sep = "")', parse = TRUE, size = 3.5) +  
annotate("text", x = replicate[4] + 0.208, y = 4, label = 'paste("[", italic(p), "= 0.001,"~"z = 3.29", "]", sep = "")', parse = TRUE, size = 3.5) +  
annotate("text", x = replicate[5] + 0.228, y = 5, label = 'paste("[", italic(p), "= 0.0001,"~"z = 3.89", "]", sep = "")', parse = TRUE, size = 3.5) +  
 guides(size = "none") +  
 xlab("Replication rate") +   
 labs(colour = NULL, y = "") +  
 theme\_bw() +   
 theme(axis.title = element\_text(size = 18),  
 axis.text.x = element\_text(size = 16),  
 axis.text.y = element\_text(size = 18),  
 plot.margin=unit(c(0,0,0,0), 'cm')) +   
 scale\_x\_continuous(limits = c(0, 1.25), breaks = seq(0, 1, by = 0.2),  
 minor\_breaks=seq(0,1,0.05), expand = expansion(mult = c(0, 0)))   
  
replicate\_typical.p



## Sample size multiplier

If the sample size of the replication study is twice as large as that of the original study, then the SNR of the replication study will be larger by a factor square root of 2. Thus, we can also compute the conditional probability of a successful replication given the z-statistic of the original study when the replication study is larger (or smaller) by some factor.

Conditional replication rates when the original study has or and the replication study is times larger.

# calculate replication probabilities when multiplying the   
# sample size of the original study with factors 1,2,...10.  
replicate\_rep\_weak = sapply(1:10, function(x) replcalc(2, p=p, m=m, s=sqrt(sigma^2-1), multiplier = x))  
replicate\_rep\_strong = sapply(1:10, function(x) replcalc(3.3, p=p, m=m, s=sqrt(sigma^2-1), multiplier = x))  
  
# prepare a data frame  
replicate\_multiplier <- data.frame(evidence = c(rep("Weak evidence", 10),   
 rep("Strong evidence", 10)),   
 mult\_values = rep(1:10, 2),   
 replicate = c(replicate\_rep\_weak,   
 replicate\_rep\_strong))  
# round  
replicate\_multiplier$replicate <- round(replicate\_multiplier$replicate, 2)  
# data frame  
replicate\_multiplier

## evidence mult\_values replicate  
## 1 Weak evidence 1 0.39  
## 2 Weak evidence 2 0.55  
## 3 Weak evidence 3 0.63  
## 4 Weak evidence 4 0.68  
## 5 Weak evidence 5 0.72  
## 6 Weak evidence 6 0.74  
## 7 Weak evidence 7 0.75  
## 8 Weak evidence 8 0.77  
## 9 Weak evidence 9 0.78  
## 10 Weak evidence 10 0.78  
## 11 Strong evidence 1 0.74  
## 12 Strong evidence 2 0.89  
## 13 Strong evidence 3 0.94  
## 14 Strong evidence 4 0.96  
## 15 Strong evidence 5 0.97  
## 16 Strong evidence 6 0.97  
## 17 Strong evidence 7 0.98  
## 18 Strong evidence 8 0.98  
## 19 Strong evidence 9 0.98  
## 20 Strong evidence 10 0.98

# figure  
replicate\_multiplier.p <- replicate\_multiplier %>%   
 ggplot(aes(x = mult\_values, y = replicate, color = evidence, fill = evidence)) +  
 geom\_point(color = "transparent") +  
 geom\_line(alpha=1, linewidth = 0.5) +   
 scale\_y\_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.2),  
 minor\_breaks=seq(0,1,0.05),   
 expand = expansion(mult = c(0, 0.01))) +  
 scale\_x\_continuous(limits = c(1, 10), breaks = seq(1, 10, by = 1),  
 labels=paste(1:10,"x",sep=''),  
 expand = expansion(mult = c(0.03, 0.03))) +   
 ylab("Replication probability") +   
 xlab(expression(paste("Relative sample size for the replication study"))) +  
 geom\_point(data = replicate\_multiplier, aes(size = replicate)) +   
 #scale\_size(range=c(8,15)) +  
 geom\_text(data = replicate\_multiplier,   
 aes(label = scales::percent(replicate)), size = 3.5, color = "gray10") +  
 guides(size = "none", color = "none", fill = "none") +  
 labs(colour = NULL) +   
 theme\_bw() +   
 theme(axis.title = element\_text(size = 18),  
 axis.text = element\_text(size = 16),  
 plot.margin=unit(c(0.2,0.3,0,0), 'cm')) +  
 annotate("text", x = 3, y = 0.5, label = "Weak evidence", size = 4) +  
 annotate("text", x = 3, y = 0.85, label = "Strong evidence", size = 4)  
replicate\_multiplier.p

