Notebook 10: free lunch

Guillaume A. Rousselet

2022-12-12

Contents

Dependencies 1

Power curves 1

Get power for different sample sizes and effect sizes . . . 2

Plot results . . . 2

Get critical effect sizes for different sample sizes 3

Plot results . . . 4

ROPE + HDI example 5

Simulate data . . . 5

Plot results . . . 6

Define ROPE . . . 7

Bootstrap analysis . . . 7

Get highest density and quantile intervals . . . 8

Plot bootstrap results . . . 8

References 9

Dependencies

```
library(tibble)
library(ggplot2)
library(pwr)
library(HDInterval)
source("./theme_gar.txt") # format ggplot2 figures
```

Power curves

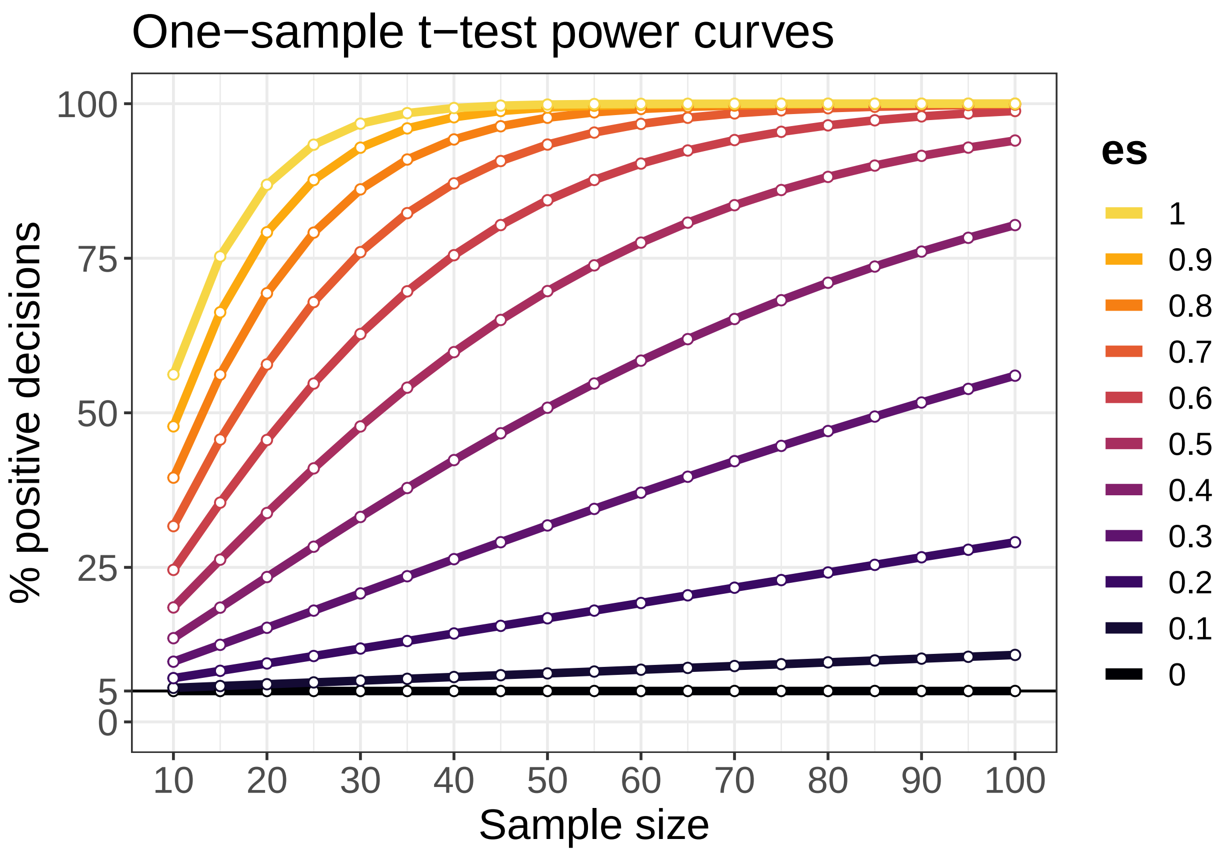
We consider the one-sample t-test only. Results are not simulated: we use parametric predictions from the `pwr` package, assuming normality.

Get power for different sample sizes and effect sizes

```r
nmax <- 100
nvec <- seq(10,nmax,5)
nn <- length(nvec)
aat <- 0.05 # arbitrary alpha threshold
esvec <- seq(0, 1, 0.1) # effect sizes
nes <- length(esvec)
# Define array of results
power.curves <- array(NA, dim = c(nn, nes))

for(N in 1:nn){
  for(ES in 1:nes){
    power.curves[N,ES] <- pwr.t.test(n = nvec[N], d = esvec[ES], sig.level = aat)$power
  }
}
```

Plot results

```r
df <- tibble(posdec = as.vector(power.curves),
             sampsize = rep(nvec, nes),
             es = factor(rep(esvec, each = nn)))

ggplot(data=df, aes(x=sampsize, y=posdec*100, colour=es)) + theme_bw() +
  geom_abline(slope = 0, intercept = 0.05*100) +
  geom_line(size=1.5) +
  geom_point(fill="white", shape=21, show.legend = FALSE) +
  scale_colour_viridis_d(option = "B", end = 0.9) +
  coord_cartesian(ylim=c(0, 100)) +
  scale_x_continuous(breaks = nvec[c(TRUE, FALSE)]) +
  scale_y_continuous(breaks = c(0, 5, seq(25,100,25))) +
  labs(x = "Sample size",
       y = "% positive decisions") +
  theme(axis.title = element_text(size = 16),
        axis.text = element_text(size = 14),
        plot.title = element_text(size = 18),
        panel.grid.minor.y = element_blank(),
        legend.title = element_text(size = 16, face = "bold"),
        legend.text = element_text(size = 12)) +
  guides(colour = guide_legend(reverse=TRUE,override.aes = list(size = 2))) +
  ggtitle('One-sample t-test power curves')
```



```
# ggsave('./figures/power_curves.png', width = 20, height = 12, units = "cm")
```

Get critical effect sizes for different sample sizes

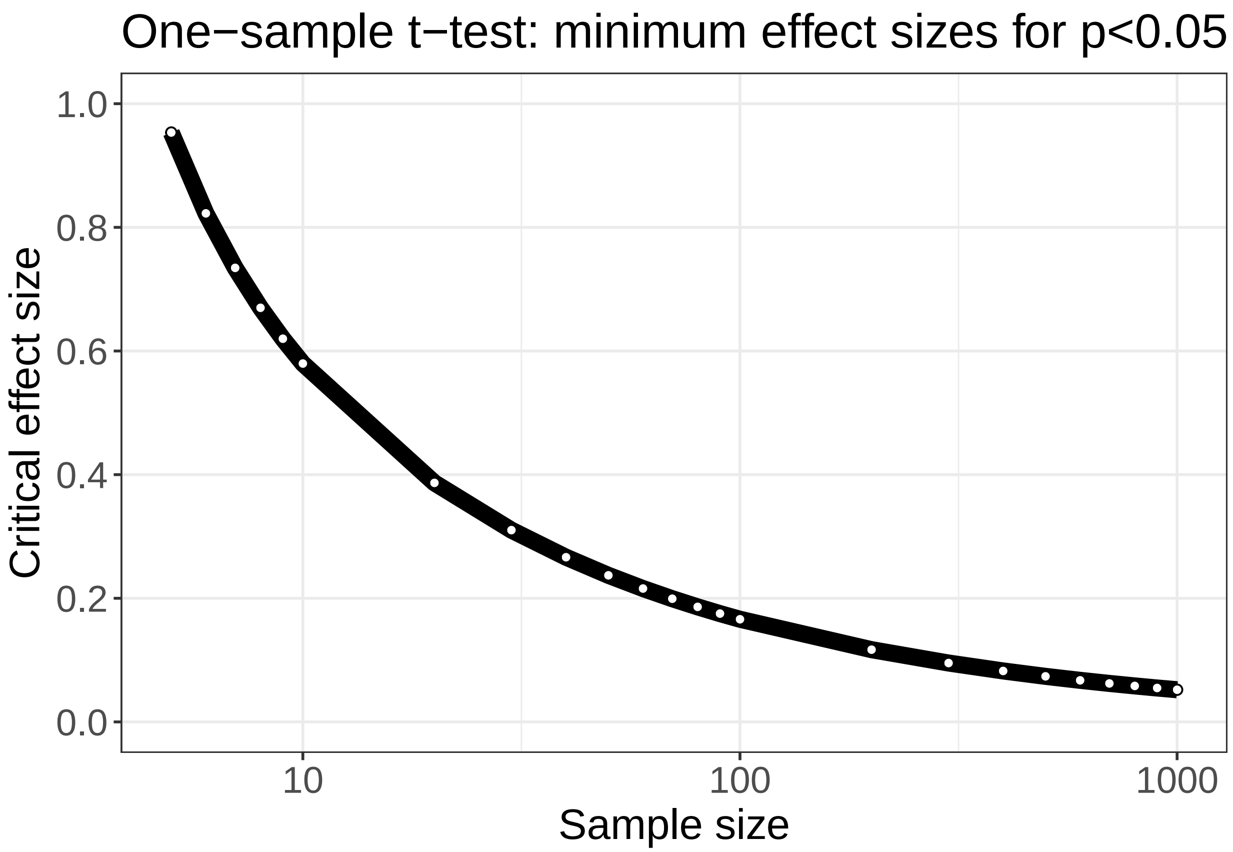
We use the formula from the one-sample t-test to determine the minimum effect size needed to reach p<0.05. A similar curve appears in figure 1 of Rouder et al. (2016).

```
nmin <- 5
nmax <- 1000
numbreaks <- 10
# nvec <- exp(seq(log(nmin), log(nmax), length.out = numbreaks))
nvec <- c(seq(5,10), seq(20,100,10), seq(200,1000,100))
nn <- length(nvec)
aat <- 0.05 # arbitrary alpha threshold
# es <- 0.4 # example effect size considered in Rouder et al. (2016)
sidt <- 1 # 1=1-sided, 2=2-sided
# Define array of results
ces.curve <- vector(mode = "numeric", length = nn)

for(N in 1:nn){
  # power <- pwr.t.test(n = nvec[N], d = es, sig.level = aat)$power
  # beta <- 1 - power

  # standard alpha
```

```r
  df <- nvec[N]-1 # degrees of freedom
  tth <- qt(1-aat/sidt, df) # threshold t value
  ces.curve[N] <- tth / sqrt(nvec[N]) # critical effect size
}
```

Plot results

```r
df <- tibble(ces = ces.curve,
             sampsize = nvec)

ggplot(data=df, aes(x=sampsize, y=ces)) + theme_bw() +
  geom_line(size=3, colour="black") +
  geom_point(fill="white", shape=21, show.legend = FALSE) +
  coord_cartesian(ylim=c(0, 1)) +
  scale_x_continuous(trans='log10') +
  scale_y_continuous(breaks = seq(0,1,0.2)) +
  labs(x = "Sample size",
       y = "Critical effect size") +
  theme(axis.title = element_text(size = 16),
        axis.text = element_text(size = 14),
        plot.title = element_text(size = 18),
        panel.grid.minor.y = element_blank(),
        legend.title = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = c(0.8, 0.8)) +
  guides(colour = guide_legend(reverse=TRUE,override.aes = list(size = 2))) +
  ggtitle('One-sample t-test: minimum effect sizes for p<0.05')
```



```
# ggsave('./figures/min_effect_size.png', width = 20, height = 12, units = "cm")
```

ROPE + HDI example

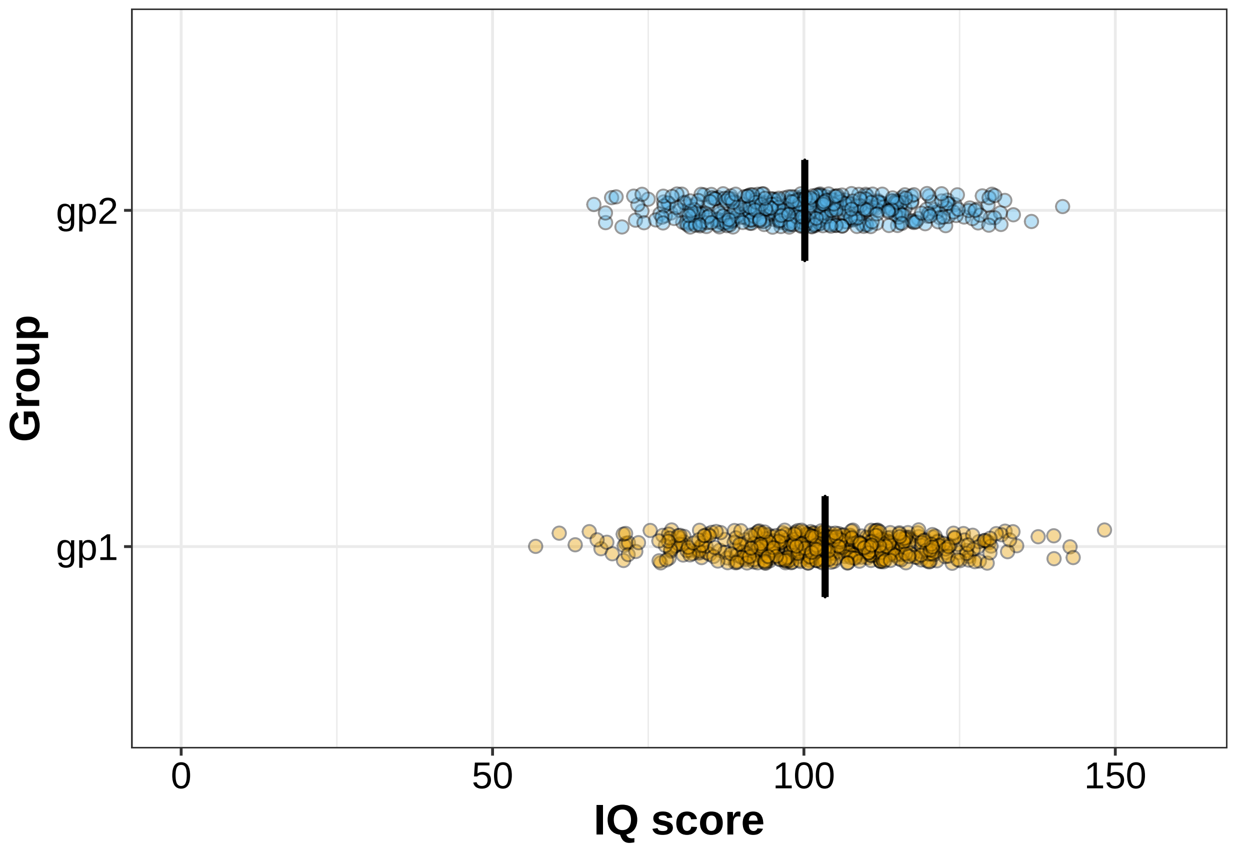
Example from Kruschke (2018). To reproduce that example, we're going to simulate the data using the modes of the posterior distributions reported in his figure 2.

Simulate data

```
set.seed(44)
# group 1
n1 <- 498
m1 <- 104
sd1 <- 14.6

# group 2
n2 <- 468
m2 <- 100
sd2 <- 14.7

gp1 <- rnorm(n = n1, mean = m1, sd = sd1)
gp2 <- rnorm(n = n2, mean = m2, sd = sd2)
```

Plot results

```
df <- tibble(IQ = c(gp1, gp2),
             Group = factor(c(rep("gp1", n1), rep("gp2", n2)))
            )

tp <- 0.05

set.seed(44) # reproducible jitter

ggplot(df, aes(x = Group,
               y = IQ,
               fill = Group)) +
geom_jitter(alpha = 0.4,
            shape = 21,
            size = 2,
            width = .05) +
scale_fill_manual(values=c("#E69F00", "#56B4E9")) + theme_gar +
  theme(legend.position = "none") +
  theme(axis.title = element_text(size=16,face="bold"),
        axis.text = element_text(size=14)) +
        ylab("IQ score") +
  # plot sample trimmed means
  stat_summary(fun = mean, fun.min = mean, fun.max = mean,
                 geom = "crossbar", width = 0.3, colour = "black",
                 fun.args = list(trim = tp)) +
  coord_flip(ylim = c(0, 160))
```



```
# ggsave(filename=('./figures/figure_iq_res.pdf'),width=7,height=4)
```

Define ROPE

In his article, Kruschke (2018) explains how to define a ROPE using a standardised weak effect size (half of 0.2). He also points out that this is a really rough strategy that should not be applied systematically. More specialist knowledge should be used to define a ROPE. But maybe this is our first exploration of an effect, so that's good enough to get us started. We could always learn from our experiment and then plan a replication study using a better ROPE.

```
# Assume the IQ population standard deviation is 15
pop.sd <- 15
# define ROPE's lower bound
rope.lo <- -0.1 * 15
# define ROPE's upper bound
rope.hi <- 0.1 * 15
```

Bootstrap analysis

As we generated the data, we know the populations are normal. But let's pretend that we expect a small amount of skewness, so we use a 5% trimmed mean as a robust estimate of central tendency.

```
set.seed(666)
nboot <- 10000 # number of bootstrap samples
tp <- 0.05 # trimming percentage
boot1 <- matrix(sample(gp1, size = n1*nboot, replace = TRUE), nrow = nboot, byrow = TRUE)
boot2 <- matrix(sample(gp2, size = n2*nboot, replace = TRUE), nrow = nboot, byrow = TRUE)
boot.diff <- apply(boot1, 1, mean, trim = tp) - apply(boot2, 1, mean, trim = tp)

samp.diff <- mean(gp1, trim = tp) - mean(gp2, trim = tp)
```

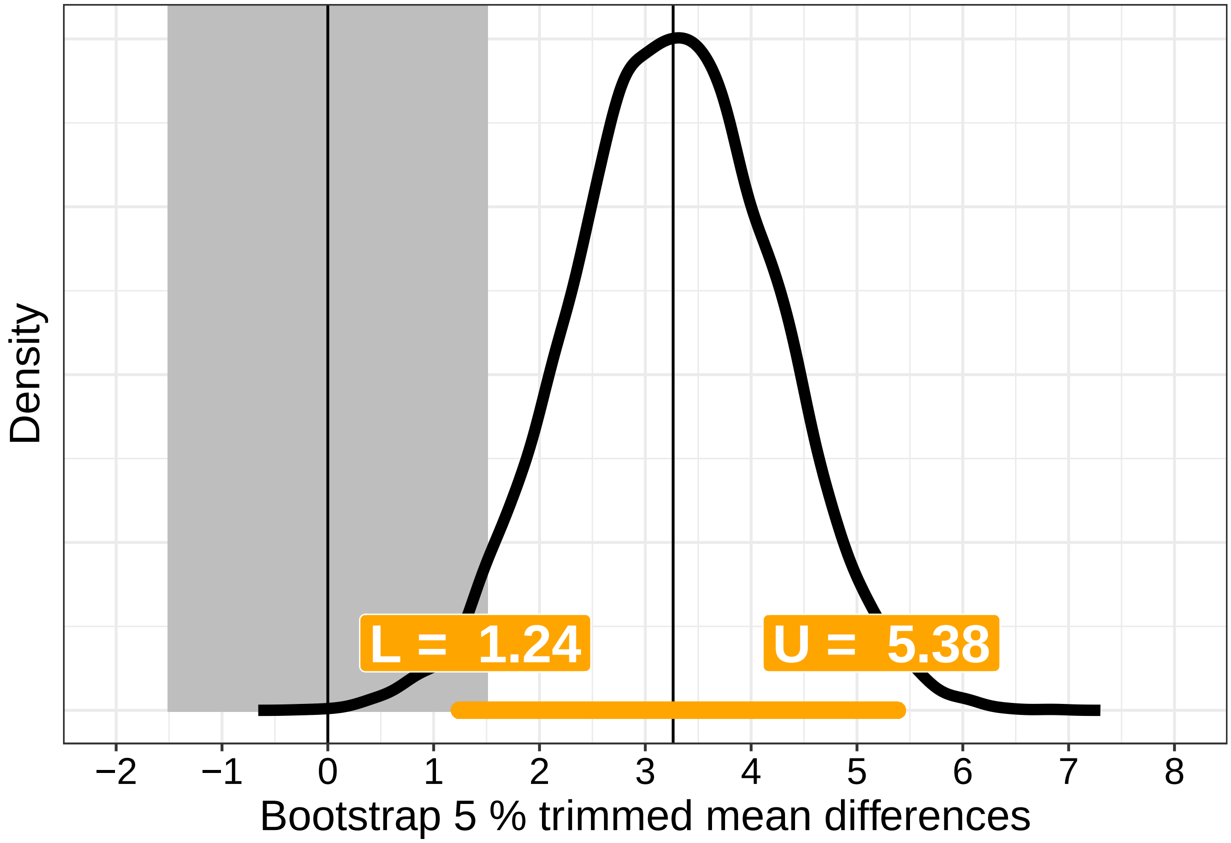
Get highest density and quantile intervals

We use 97% intervals, because 97 is a prime number, and it is as a good a justification as using an arbitrary 95%.

```
alpha <- 0.03
boot.ci <- quantile(boot.diff, probs = c(alpha/2, 1-alpha/2))
boot.hdi <- HDInterval::hdi(boot.diff, credMass = 1-alpha)
```

Plot bootstrap results

```
# choose ci or hdi
# ci <- boot.ci
ci <- boot.hdi

df <- as_tibble(with(density(boot.diff),data.frame(x,y)))

ggplot(df, aes(x = x, y = y)) + theme_gar +
  # plot ROPE
  geom_rect(aes(xmin=rope.lo, xmax=rope.hi, ymin=0, ymax=Inf), colour = "grey", fill = "grey") +
  # geom_vline(xintercept = rope.lo, colour = "grey", size = 1) +
  # geom_vline(xintercept = rope.hi, colour = "grey", size = 1) +
  # plot zero reference line
  geom_vline(xintercept = 0, colour = "black", size = 0.5) +
  # plot bootstrap distribution
  geom_line(size = 2) +
  # plot sample difference
  geom_vline(xintercept = samp.diff, colour = "black", size = 0.5) +
  scale_x_continuous(breaks = seq(-10, 10, 1)) +
  coord_cartesian(xlim = c(-2, 8)) +
  labs(x = paste("Bootstrap",tp*100,"% trimmed mean differences"), y = "Density") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank()) +
  # plot confidence interval ----------------------
  geom_segment(x = ci[1], xend = ci[2],
               y = 0, yend = 0,
               lineend = "round", size = 3, colour = "orange") +
  annotate(geom = "label", x = ci[1]+0.15, y = 0.1*max(df$y), size = 7,
             colour = "white", fill = "orange", fontface = "bold",
             label = paste("L = ", round(ci[1], digits = 2))) +
  annotate(geom = "label", x = ci[2]-0.15, y = 0.1*max(df$y), size = 7,
             colour = "white", fill = "orange", fontface = "bold",
             label = paste("U = ", round(ci[2], digits = 2)))
```



```
# ggsave(filename=('./figures/figure_boot_rope_hdi.pdf'),width=7,height=4)
```

References

Kruschke (2018) Rejecting or Accepting Parameter Values in Bayesian Estimation. Advances in Methods and Practices in Psychological Science 1(2):270-280. doi:10.1177/2515245918771304 https://journals.sagepub.com/doi/10.1177/2515245918771304

Rouder, J.N., Morey, R.D., Verhagen, J., Province, J.M. and Wagenmakers, E.-J. (2016), Is There a Free Lunch in Inference?. Top Cogn Sci, 8: 520-547. https://doi.org/10.1111/tops.12214