Bootstrapping & Resampling Methods

general problem

- scientific Qs are about populations
- we can’t measure entire populations
- experiments generate samples
- samples -> estimate population parameters
- “parametric” approaches come with assumptions

general problem

- what if assumptions are violated?
- data are not normally distributed
- variances unequal
- sample size unequal
- nonlinear model
- etc etc

bootstrapping

1. a way to estimate the precision of sample-based population estimates (without having access to the entire population)
   - doesn’t rely on parametric assumptions (e.g. normality)
2. a way to do hypothesis testing
   - non-parametric, by simulating the null
3. a way to do power calculations
   - not restricted by assumptions

1. Estimating Population Parameters

- we saw earlier:
  - best estimate of a population mean is the sample mean (assuming normality)

    $$\hat{\mu} = \bar{X} = \frac{\sum X_i}{N}$$

  - estimate of sd of sampling distribution of means is standard error of mean:

    $$s_{\bar{x}} = \frac{s_x}{\sqrt{N}}$$

  - can use this to generate 95% CIs of population mean

    $$\bar{X} \pm t_{\alpha}(s_{\bar{x}})$$

1. Estimating Population Parameters

- bootstrapping can estimate sampling distribution of means
- no need to assume any particular theoretical distribution
- use resampling with replacement to simulate repeatedly sampling from the population
- uses sample as proxy for population

1. Estimating Population Parameters

assume you have a sample X1…Xn and a statistic of interest (e.g. the mean)

```
repeat M times (where M is large, e.g. 10,000)
    generate a new sample of size n by resampling, with replacement, from X1..Xn
    compute the statistic based on the new sample
    set that statistic aside (e.g. save it in a list)
```

now you have a list of M versions of the statistic, one for each resampling

that list represents an **empirical bootstrap distribution of the statistic of interest**

now you can compute relevant quantities of that distribution (e.g. 95% CIs)

1. Estimating Population Parameters

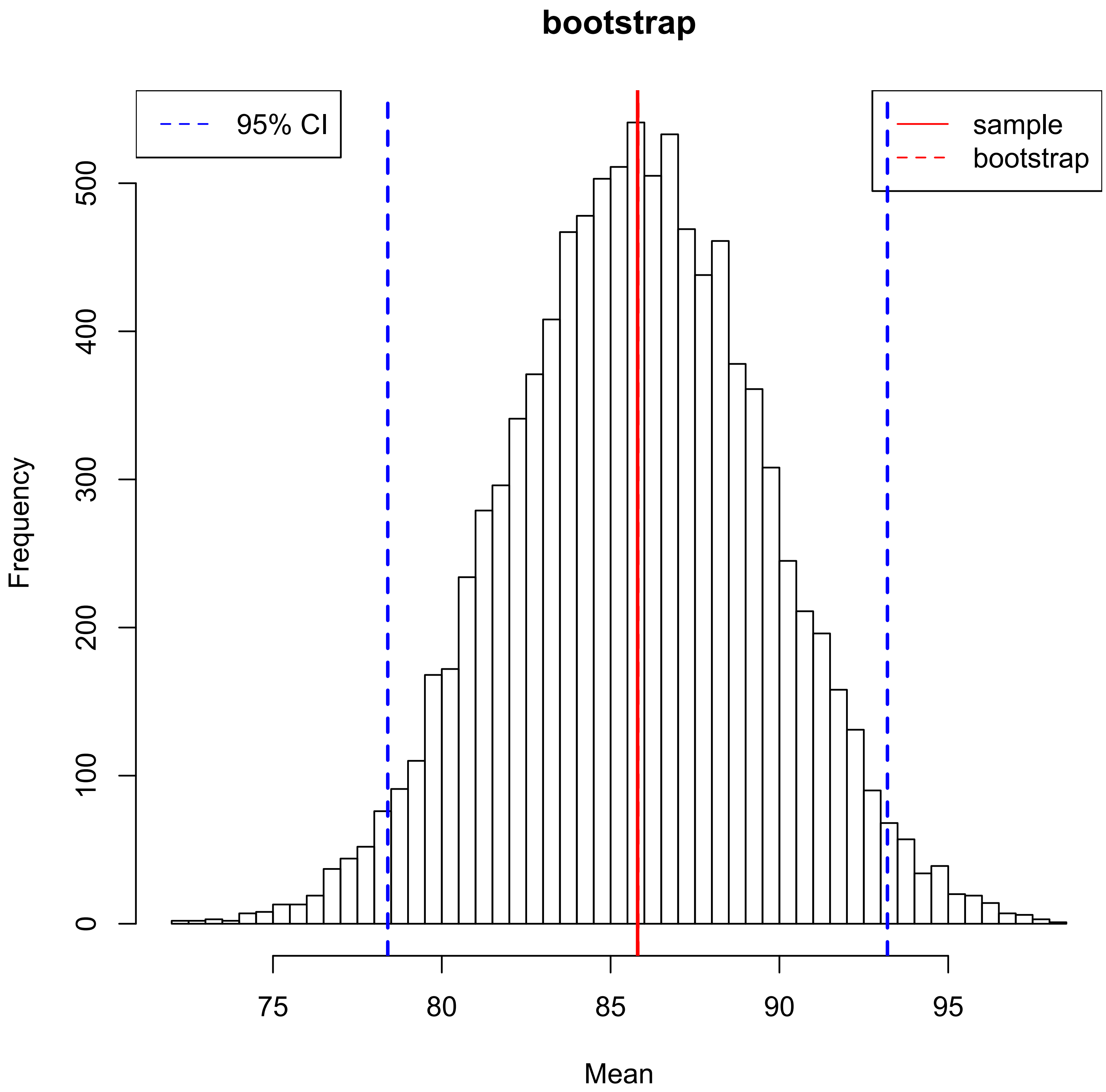
- e.g. we have a sample of size 20:
- 66 79 93 86 69 79 101 97 91 95
  72 106 105 75 70 85 92 74 88 93
- estimate of population mean (using sample mean) is 85.8
- how precise is that estimate?

1. Estimating Population Parameters

```
X = c(66, 79, 93, 86, 69, 79, 101, 97, 91, 95, 72, 106, 105, 75, 70, 85, 92, 74, 88, 93)

# compute a statistic of interest
(Xm = mean(X))

# use resampling to generate an empirical bootstrap distribution of that statistic

# how many simulated experiments?
boot_m = 10000

# create a list to store our bootstrap values
Xm_boot = array(NA, boot_m)

# do it
for (i in 1:10000) {
    Xb = sample(X, length(X), replace=TRUE) # generate new sample
    Xm_boot[i] = mean(Xb)            # compute statistic of interest
}

# display results
hist(Xm_boot, xlab="Mean", main="bootstrap")
abline(v=Xm, col="red")
abline(v=mean(Xm_boot), col="red", lty=2)
legend(x="topright", lty=c(1,2), col=c("red","red"), legend=c("sample","bootstrap"))

# compute 95% CI
(CI95 = quantile(Xm_boot, probs=c(.025,.975)))
abline(v=CI95[1], lty=2, col="blue")
abline(v=CI95[2], lty=2, col="blue")
legend(x="topleft", lty=2, col="blue", legend="95% CI")
```

www.gribblelab.org/stats/exercises/S10code.R

1. Estimating Population Parameters



1. Estimating Population Parameters

- here we used a bootstrap to estimate the sampling distribution of the mean
- we can do the same procedure to estimate the sampling distribution of **any statistic** we want
- e.g. variance, or median, or skew, …
- or anything we make up
- bootstrapping will estimate sampling distribution

2. Hypothesis Testing

- example: comparing two populations
- drug vs control
- null hypothesis: drug has no effect
    - drug & control **sampled from same population**
- alternate hypothesis: drug has an effect
    - drug & control not sampled from same population

2. Hypothesis Testing

- choose a test statistic (e.g. the difference between means… but could be anything; t, F, sd, whatever your scientific question calls for)
- do many many times (e.g. 10,000):
  - simulate the null hypothesis (that drug & control are sampled from same population)
- how many times did you get a test statistic as large or larger as the original one? < 5%? then reject H0

2. Hypothesis Testing

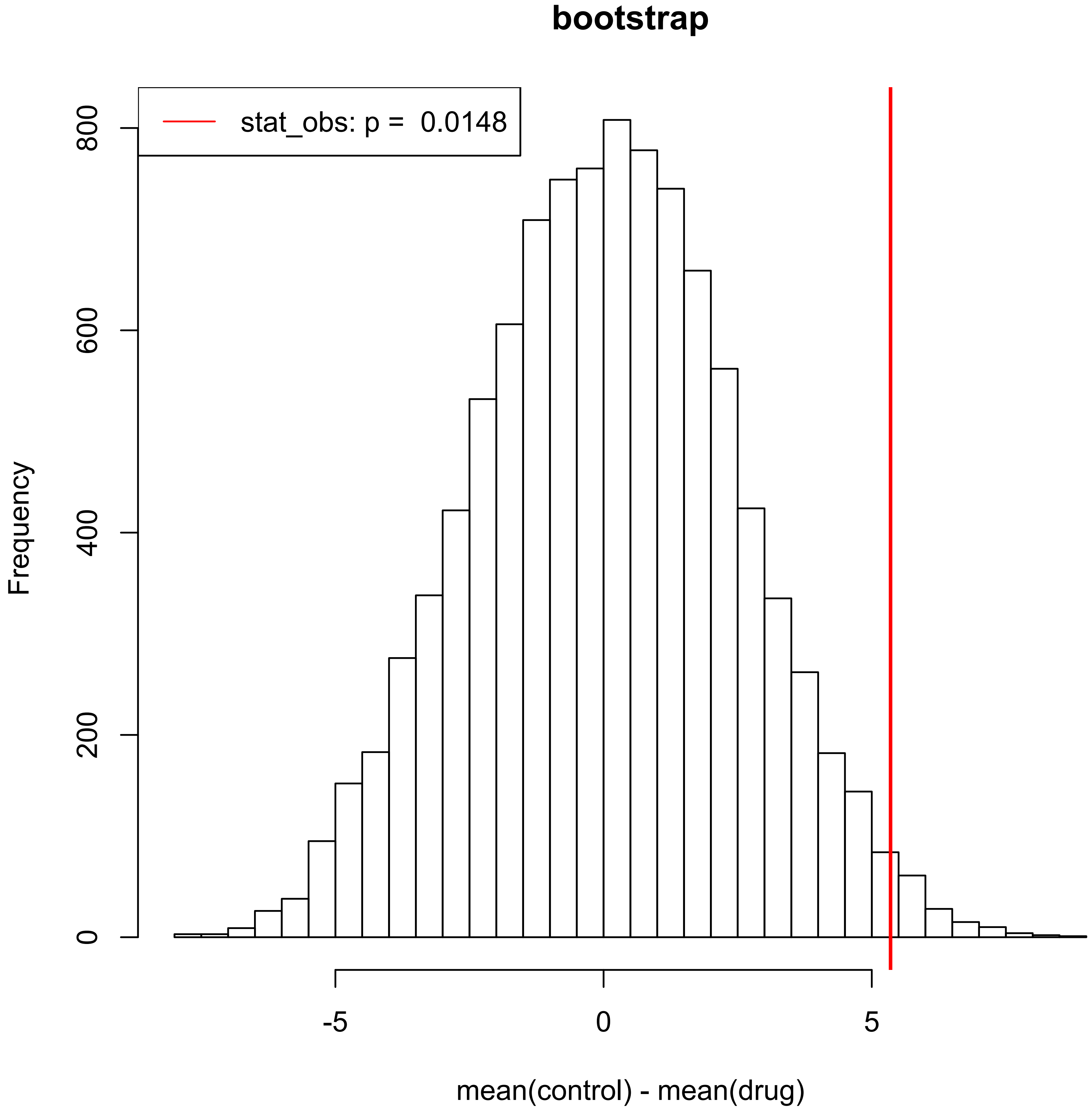
- choose a test statistic (e.g. the difference between means… but could be anything; t, F, sd, whatever your scientific question calls for)
- do many many times (e.g. 10,000):
    - throw both groups into a bucket
    - randomly reconstitute the two groups, disregarding their original group membership
    - recompute the statistic of interest
- how many times did you get a test statistic as large or larger as the original one? < 5%? then reject H0

2. Hypothesis Testing

```r
# our control group
g_control <- c(87,90,82,77,71,81,77,79,84,86,78,84,86,69,81,75,70,76,75,93)

# our drug group
g_drug <- c(74,67,81,61,64,75,81,81,81,67,72,78,83,85,56,78,77,80,79,74)

# our statistic of interest here is the difference between means
(stat_obs <- mean(g_control) - mean(g_drug))

# how many simulated experiments?
n_boot = 10000

# create a list to store our bootstrap values
stat_boot = array(NA, n_boot)

# now do a bootstrap to simulate the null hypothesis,
# namely that both groups were sampled from the same population
n_c = length(g_control)
n_d = length(g_drug)
g_bucket = c(g_control, g_drug)
for (i in 1:n_boot) {
    # reconstitute both groups, ignoring original labels
    permuted_order <- sample(1:(n_c+n_d), n_c+n_d, replace=FALSE)
    permuted_bucket <- g_bucket[permuted_order]
    boot_control <- permuted_bucket[1:n_c]
    boot_drug <- permuted_bucket[(n_c+1):(n_c+n_d)]
    stat_boot[i] <- mean(boot_control) - mean(boot_drug)
}

# visualize the empirical bootstrap distribution of our statistic of interest
hist(stat_boot, xlab="mean(control) - mean(drug)", main="bootstrap")
abline(v=stat_obs, col="red")

# how many times in the bootstrap did we observe a stat_boot as big or bigger than our stat_obs?
(p_boot <- length(which(stat_boot >= stat_obs)) / n_boot)
legend(x="topleft", lty=1, col="red", legend=paste("stat_obs: p = ", p_boot))
```

www.gribblelab.org/stats/exercises/S10code.R

2. Hypothesis Testing



2. Hypothesis Testing

- here we tested the difference between means
- but we can apply this method to any statistic of interest that we can calculate
- no need to assume theoretical distribution
- compute probability under H0 empirically by simulating the null hypothesis

3. Power Calculations

- we can use random resampling to simulate experiments not only under the null hypothesis but under any alternate hypothesis of our choosing
- we can use simulations to answer questions about statistical power

3. Power Calculations

- what's the probability of detecting a given effect with a given number of subjects?
- how many subjects are required to detect a given effect 80% of the time? (or any other % of your choosing)
- again a bootstrapping/resampling approach doesn't require assumptions about a theoretical distribution

3. Power Calculations

- example: 2 groups, drug and control
- control
  87 90 82 87 71 81 77 79 84 86
  78 84 86 69 81 75 70 76 75 93
- drug
  74 73 81 65 64 75 76 81 81 67
  72 78 83 75 66 78 77 80 79 74
- Mann-Whitney U test:
  t = 2.0613
  p = 0.04626

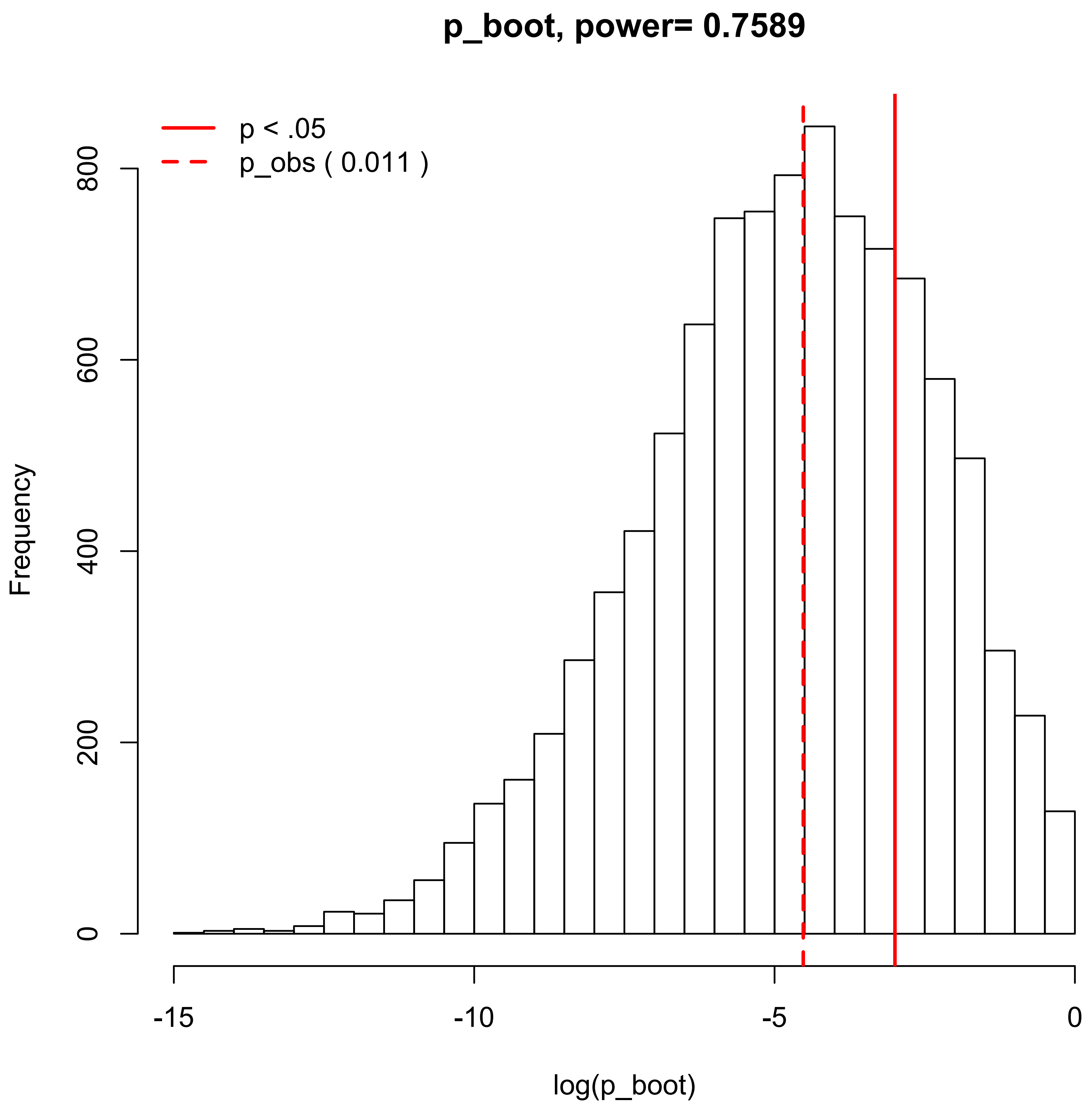
what is our statistical power?

3. Power Calculations

```r
# our two groups
g_control <- c(87,90,82,87,71,81,77,79,84,86,78,84,86,69,81,75,70,76,75,93)
g_drug <- c(74,73,81,65,64,72,76,81,81,67,72,78,83,75,66,78,77,80,79,74)

# do a Mann-Whitney U test (nonparametric version of a t-test)
out <- wilcox.test(g_control, g_drug)
w_obs <- out$statistic
p_obs <- out$p.value

n_boot <- 10000
w_boot = array(NA, n_boot)
p_boot = array(NA, n_boot)
for (i in 1:n_boot) {
    b_control <- sample(g_control,length(g_control),replace=TRUE)
    b_drug <- sample(g_drug,length(g_drug),replace=TRUE)
    out <- wilcox.test(b_control, b_drug)
    w_boot[i] <- out$statistic
    p_boot[i] <- out$p.value
}

(power <- length(which(p_boot <= .05)) / n_boot)

hist(log(p_boot), 100, main=paste("p_boot, power=", power), xlab="p_boot")
abline(v=log(0.05), col="red", lty=1, lwd=2)
abline(v=log(p_obs), col="red", lty=2, lwd=2)
legend(x="topleft", col="red", lty=c(1,2), lwd=2, legend=c("p < .05", paste("p_obs (",round(p_obs,3),")")), box.lty=0)
```

www.gribblelab.org/stats/exercises/S10code.R

3. Power Calculations



3. Power Calculations

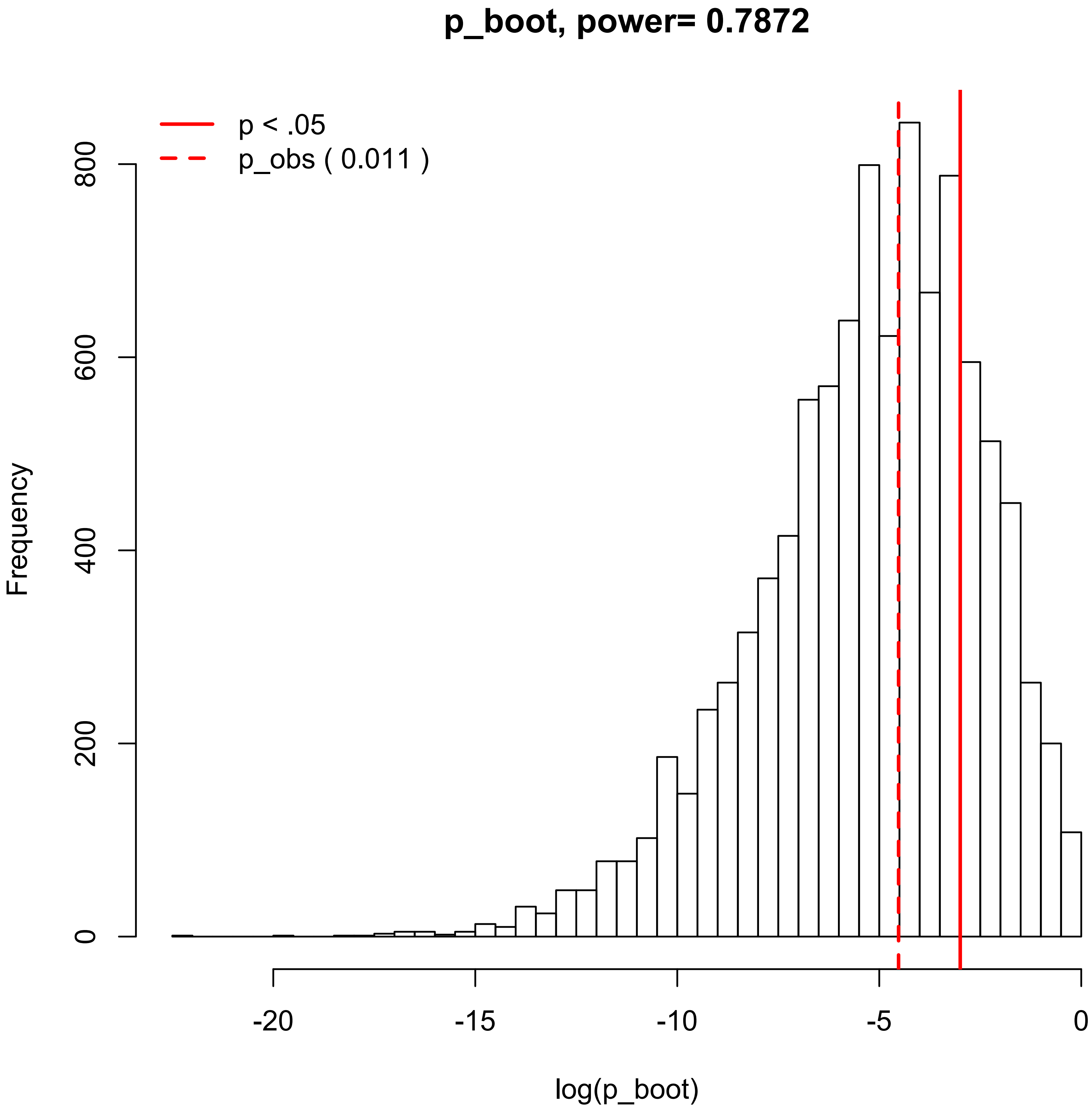
- here we used bootstrap to simulate re-doing an experiment many times
- we used a Mann-Whitney U test as our statistical test
- but one could use anything (e.g. a t-test)
- If you are OK with assuming a theoretical distribution (e.g. a t distribution) then you can perform a **parametric bootstrap**

3. Power Calculations

```r
n_control <- length(g_control)
m_control <- mean(g_control)
sd_control <- sd(g_control)
n_drug <- length(g_drug)
m_drug <- mean(g_drug)
sd_drug <- sd(g_drug)

for (i in 1:n_boot){
    b_control <- rnorm(n_control, mean=m_control, sd=sd_control)
    b_drug <- rnorm(n_drug, mean=m_drug, sd=sd_drug)
    out <- wilcox.test(b_control, b_drug)
    w_boot[i] <- out$statistic
    p_boot[i] <- out$p.value
}

(power <- length(which(p_boot <= .05)) / n_boot)

hist(log(p_boot), 50, main=paste("p_boot, power=", power), xlab="log(p_boot)")
abline(v=log(0.05), col="red", lty=1, lwd=2)
abline(v=log(p_obs), col="red", lty=2, lwd=2)
legend(x="topleft", col="red", lty=c(1,2), lwd=2, legend=c("p < .05", paste("p_obs (",round(p_obs,3),")")), box.lty=0)
```

www.gribblelab.org/stats/exercises/S10code.R

3. Power Calculations



3. Power Calculations

- in a parametric bootstrap instead of simulating the experiment by resampling from your sample,
- instead you sample from the best estimate of the population distribution
- e.g. for the previous example, if we're ok to assume a normal distribution, then
- control: Normal(mean=80.55, sd=6.70)
  drug: Normal(mean=74.8, sd=5.74)

non-parametric statistical tests

- unpaired t-test: Mann-Whitney U test
- paired t-test: Wilcoxon test
- one factor ANOVA: Kruskal-Wallis test
- correlation: Spearman rank-order correlation
- etc etc