Introduction to bootstrap

Tuan V. Nguyen

Genetics Epidemiology of Osteoporosis Lab

Garvan Institute of Medical Research

Study 1

Transit times (hr) of marker pellets through the alimentary canal of patients with diverticulosis on 2 treatments

Treatment A: 44, 51, 52, 55, 60, 62, 66, 68, 69, 71, 71, 76, 82, 91, 108

Treatment B: 52, 64, 68, 74, 79, 83, 84, 88, 95, 97, 101, 116

Is there difference between the two treatments beyond chance fluctuation?

Solution: Student's t-test

Determine:

- mean of group 1 and group 2 (x_1 and x_2)
- mean difference

$$d = x_1 - x_2$$

- variance → standard error (se) of d
- **t-test**

$$t = d \,/\, \mathrm{se}(d)$$

Assumptions of t-test

- Individuals from both groups are chosen randomly
- Data are *normally* distributed
- Two groups are *independent*
- Two groups have equal variance (homogeneity)

What to do when assumptions not met?

- Transformation of data
- Non-parametric test
  - Express the original data in ranks
  - Do the analysis based on ranks

Mann-Whitney-Wilcoxon test

Mann-Whitney-Wilcoxon U test

```
TreatA = c(44, 51, 52, 55, 60, 62, 66, 68, 69, 71,
  71, 76, 82, 91, 108)
TreatB = c(52, 64, 68, 74, 79, 83, 84, 88, 95, 97,
  101, 116)

Time = c(TreatA, TreatB)

Treatment = c(rep("A", 15), rep("B", 12))
```

```
> Time
 [1]  44  51  52  55  60  62  66  68  69  71  71  76  82  91 108  52  64
[18]  68  74  79  83  84  88  95  97 101 116
> rank(Time)
 [1]  1.0  2.0  3.5  5.0  6.0  7.0  9.0 10.5 12.0 13.5 13.5 16.0 18.0
[14] 22.0 26.0  3.5  8.0 10.5 15.0 17.0 19.0 20.0 21.0 23.0 24.0 25.0
[27] 27.0
```

Mann-Whitney-Wilcoxon test using R

```
> wilcox.test(Time ~ Treatment)

        Wilcoxon rank sum test with continuity correction

data:  Time by Treatment
W = 45, p-value = 0.02983
alternative hypothesis: true location shift is not equal to 0
```

Problem with non-parametric test

- Only provides P value
- No confidence interval of the difference
- Low power (chance) to detect a real effect

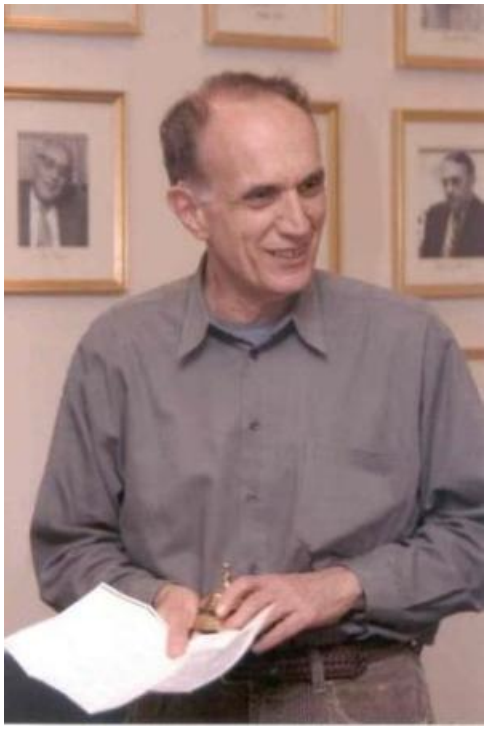
Introducing bootstrap

What to do when …

- Want to estimate 95% confidence interval for *median, variance, standard deviation, regression coefficients, proportion, ratio*
- Methods not available in classical statistics
- Sample size is small

Solution: bootstrap idea

- Professor Bradley Efron (Stanford University), 1979
- A revolution in statistical science
- Based on the idea of **repeated sampling**



Sampling

- Selection of subset (or sample) from a population
- Many ways to draw random samples: simple random sampling, systematic sampling, multistage sampling, cluster sampling, etc
- Sampling plays a central role in statistical inference

Two types of sampling

- **Sampling <u>without</u> replacement**: an element can be selected only one time

```
sample(x, n, replace=F)
```

- **Sampling <u>with</u> replacement**: an element can be selected more than one time

```
sample(x, n, replace=T)
```

Example of sampling without replacement

```
x = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

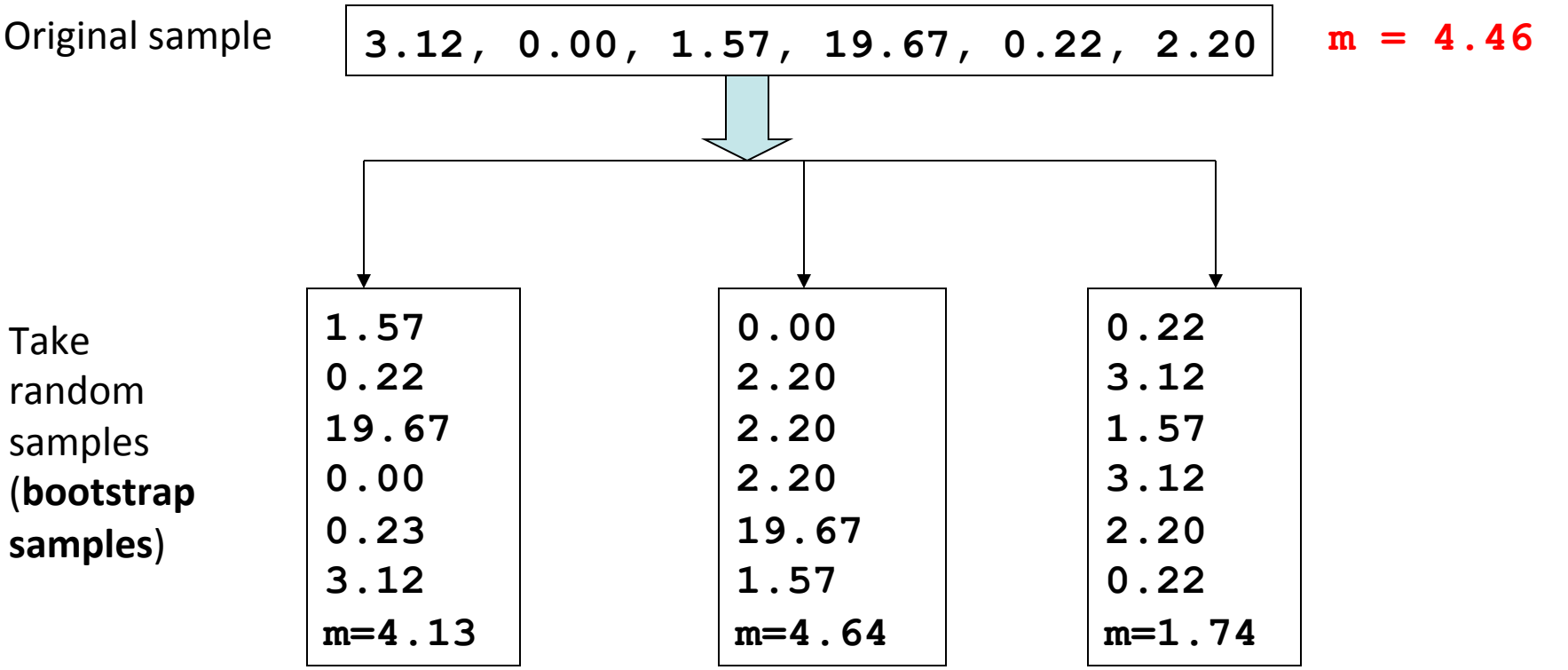
Take 5 values

```
> sample(x, 5)
[1] 4 7 2 9 8

> sample(x, 5)
[1] 9 7 2 5 4

> sample(x, 5)
[1] 9 8 6 5 4

> sample(x, 5)
[1]  7  1  8 10  9

> sample(x, 5)
[1]  9 10  1  4  2
```

Example of sampling with replacement

```
x = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

> sample(x, 5, replace=T)
[1]  7  8 10  4  1

> sample(x, 5, replace=T)
[1] 8 3 6 2 9

> sample(x, 5, replace=T)
[1] 10  1  3  5 10

> sample(x, 5, replace=T)
[1] 5 4 8 1 5

> sample(x, 5, replace=T)
[1] 5 3 5 5 5

> sample(x, 5, replace=T)
[1] 10  6  3  7  8
```

Bootstrap samples



Using R

```
original.sample = c(3.12, 0.00, 1.57, 19.67, 0.22, 2.20)
bs.sample = sample(original.sample, replace=T)
bs.sample
```

Bootstrap idea

- Step 1: start with the original sample (x_1, x_2, x_3, …, x_n);
- Step 2: take random sample with replacement → (x_1, x_1, x_2, x_4…) and calculate the statistic of interest, called it *t*;
- Repeat step 2 for B times (B can be 10000)

  (x_1, x_1, x_2, x_4…) → t_1

  (x_1, x_1, x_2, x_4…) → t_2

  (x_1, x_1, x_2, x_4…) → t_3

  …

  (x_1, x_1, x_2, x_4…) → t_B
- Collect all sample values of t
- Examine the distribution of t

Finding 95% CI for a median

- Consider the sample data

```
original.sample = c(3.12, 0.00, 1.57, 19.67, 0.22, 2.20)
```

- Want to find out the 95% confidence interval (CI) for the median
- There is no method for determining the CI for a median

Finding 95% CI for a median: bootstrap

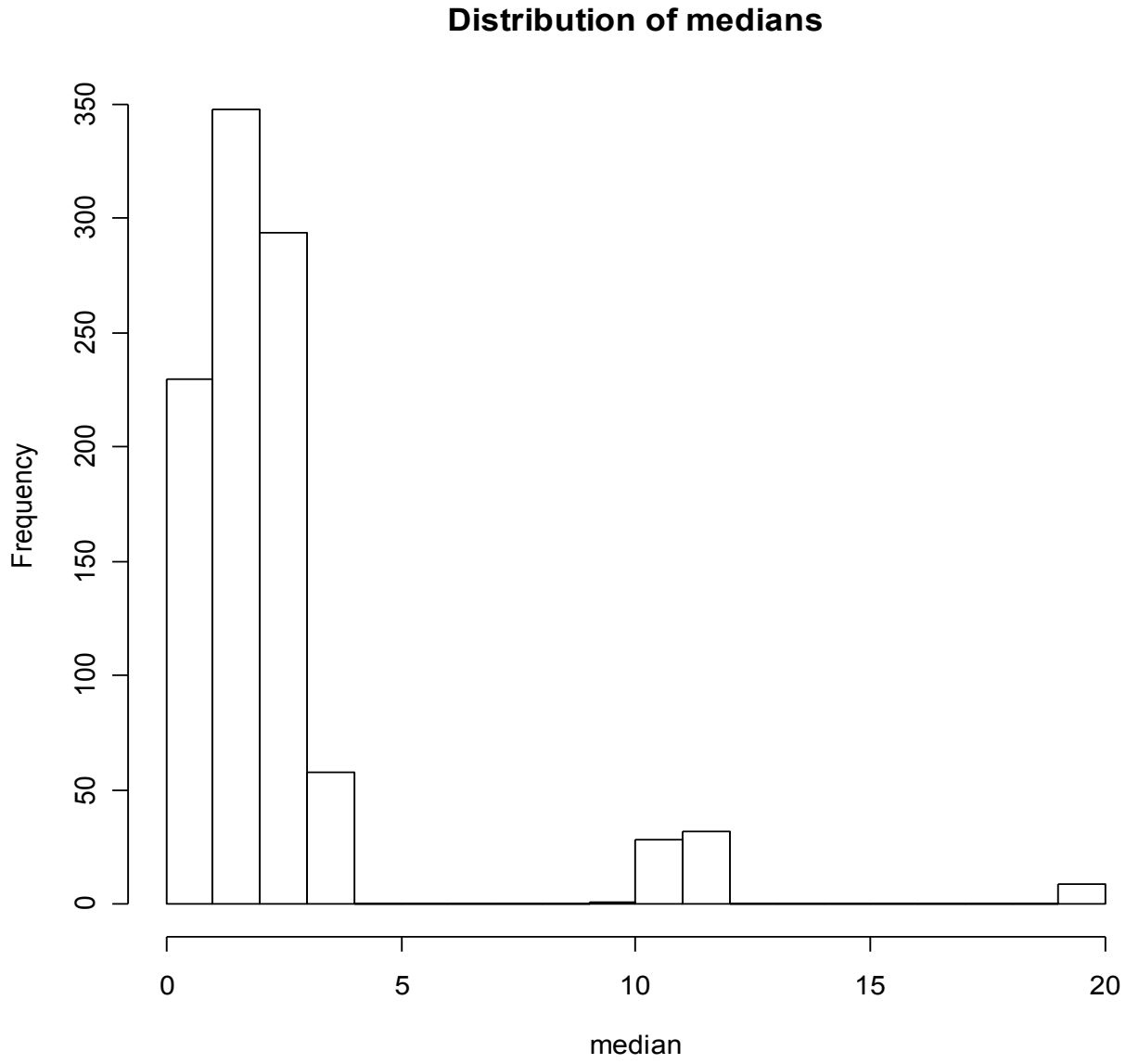
- Bootstrap solution
- Step 1: take a random sample from the original sample, calculate median (M)
- Step 2: repeat step 1 for B times
- Step 3: we now have B medians (M_i, i = 1, 2, 3, …, B)
- Step 4: examine the distribution of M_i

R implementation

```
# original sample
original.sample = c(3.12, 0.00, 1.57, 19.67, 0.22, 2.20)
n = length(original.sample)
# number of bootstrap samples = 1000
B = 1000
# create an empty vector
median = numeric(B)
# take samples and calculate the median in each sample
for (i in 1:B)
  {
    bs.sample = sample(original.sample, n, replace=T)
    median[i] = median(bs.sample)
 }

# get a histogram of the medians
hist(median, breaks=20, main="Distribution of medians")
# get median and 95% CI
quantile(median, probs=c(0.025, 0.975))
```

Notes: the above programming can be done more efficiently

```
original.sample = c(3.12, 0.00, 1.57, 19.67, 0.22, 2.20)
N = length(original.sample)
B = 1000
median = c()
for (i in 1:B)
median = c(median, median(sample(original.sample, N,
replace=T)))
quantile(median, c(0.025, 0.50, 0.975))
```




```
> quantile(median, probs=c(0.025, 0.50, 0.975))
   2.5%    50%  97.5%
 0.110  1.885 11.395
```

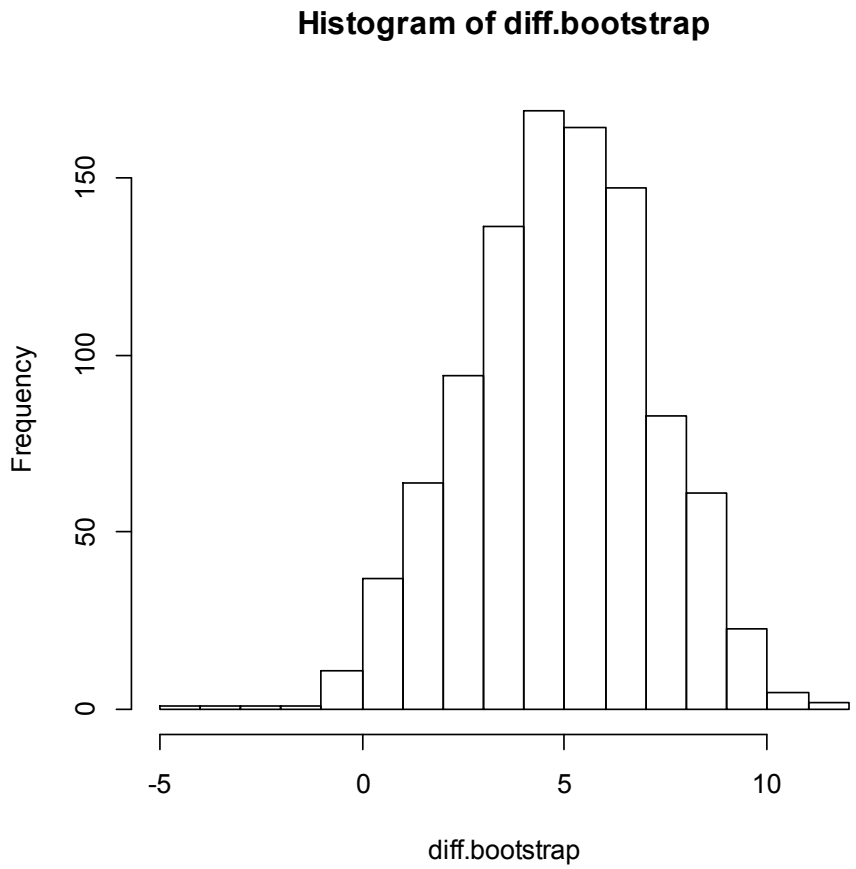
Bootstrap with R

```
# Heterozygous (BA)
a = c(86, 88, 89, 89, 92, 93, 94, 94, 94, 95, 95, 96, 96, 97,
97, 98, 98, 99, 99, 101, 106, 107, 110, 113, 116, 118)
# Homozygous (BB)
b = c(89, 90, 92, 93, 93, 96, 99, 99, 99, 102, 103, 104, 105,
106, 106, 107, 108, 108, 110, 110, 112, 114, 116, 116)
# Difference between means of observed datasets
diff.observed = mean(b) - mean(a)

# Level of significance
alpha = 0.05
```

Bootstrap with R

```
# Number of replicates
n = 1000

# Difference between means of bootstrapped datasets (n
replicates)

diff.bootstrap = NULL

for (i in 1 : n) {
    # Sample with replacement
    a.bootstrap = sample  (a, length(a), TRUE)
    b.bootstrap = sample  (b, length(b), TRUE)

    diff.bootstrap[i] = mean(b.bootstrap) - mean(a.bootstrap)
}

# Confidence interval
quantile(diff.bootstrap, c(alpha/2, 0.50, 1-alpha/2))
```

```
> quantile(diff.bootstrap, c(alpha/2, 0.50, 1-alpha/2))
     2.5%       50%     97.5%
0.3967147 4.9535256 9.1925481
> hist(diff.bootstrap, breaks=20)
```



Applications of bootstrap

- Bootstrap method can be applied to
  - Test for difference between 2 groups
  - Estimate correlation coefficient
  - Ratio of two random variables
  - and many more …

Comparison of two groups: unsual data

- Two groups of dementia patients
- Outcome: daily activity score
- Question: Is there an effect of treatment

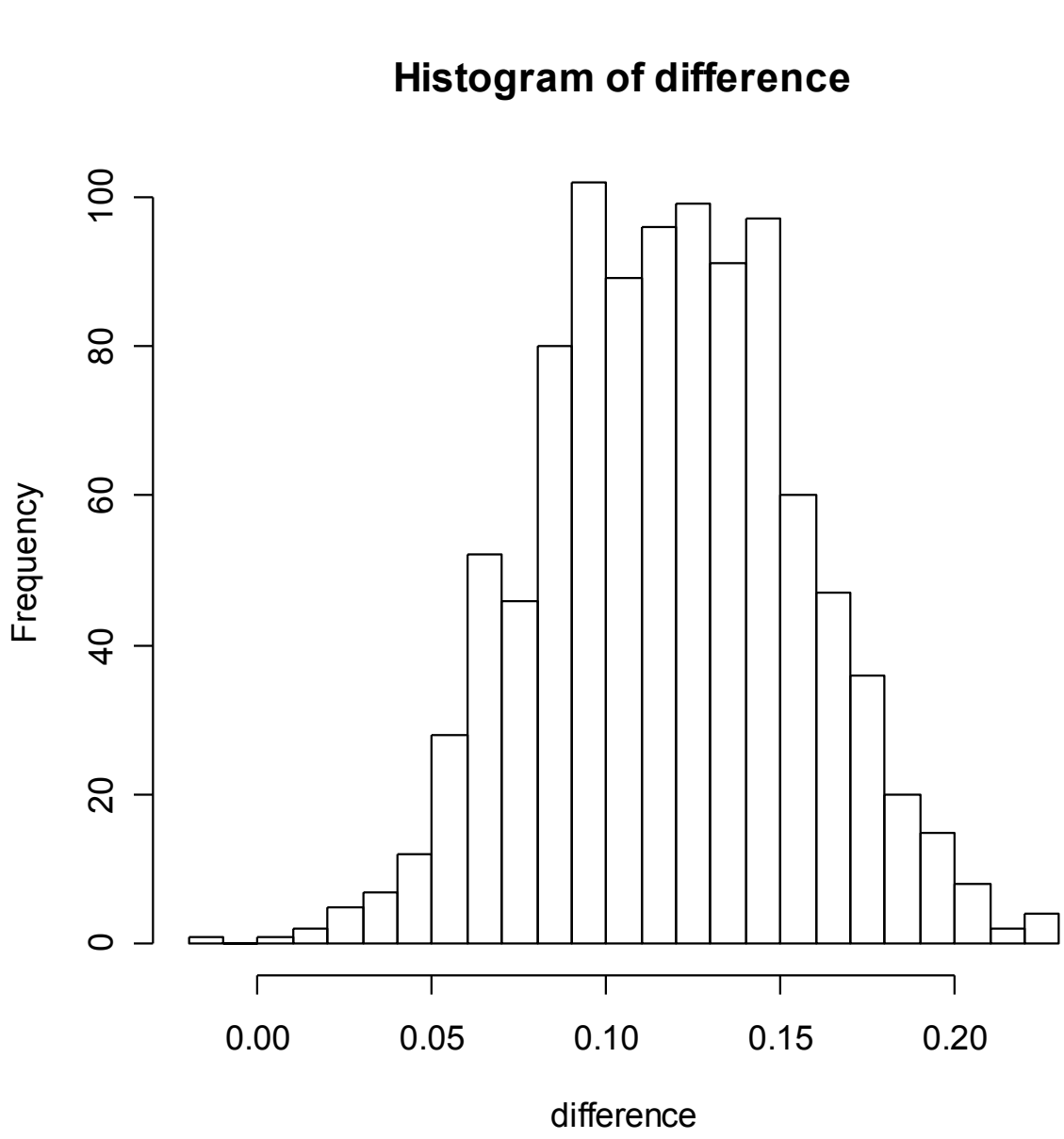
Treated	Placebo
0.05	0
0.15	0.15
0.35	0
0.25	0.05
0.20	0
0.05	0
0.10	0.05
0.05	0.10
0.30	
0.05	
0.25	
N 11	8
Mean 0.164	0.044
SD 0.112	0.056

Bootstrap solution

- Take random sample from group 1, calculate mean (m_1); take random sample from group 2, calculate mean (m_2); calculate difference $d = m_1 - m_2$
- Repeat the sampling for B times, and we now have B values of d
- Examine the distribution of d

R implementation

```
treated = c(0.05, 0.15, 0.35, 0.25, 0.20, 0.05, 0.10, 0.05,
              0.30, 0.05, 0.25)
control = c(0, 0.15, 0, 0.05, 0, 0, 0.05, 0.10)
n1 = length(treated)
n2 = length(control)
B = 1000
difference = numeric(B)
no.effect = 0
for (i in 1:B) {
   bs.treated = sample(treated, n1, replace=T)
   bs.control = sample(control, n2, replace=T)
   difference[i] = mean(bs.treated) - mean(bs.control)
   if (difference[i] <= 0) no.effect = no.effect+1
  }

hist(difference, breaks=20)
quantile(difference, probs=c(0.025, 0.50, 0.975))
no.effect/1000
```


Histogram of difference
Frequency
0
20
40
60
80
100
0.00
0.05
0.10
0.15
0.20
difference

```
> quantile(difference, probs=c(0.025, 0.50, 0.975))
      2.5%         50%       97.5%
0.04943182 0.11818182 0.19092330

> no.effect/1000
[1] 0
```

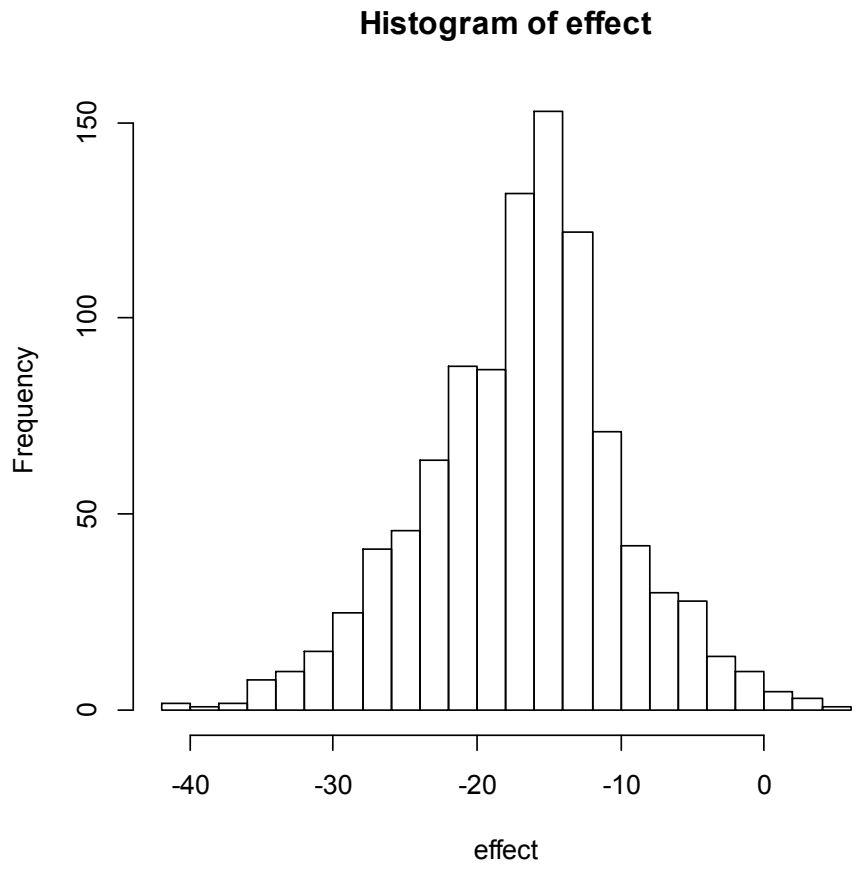
Comparing with classical method

```
> t.test(treated,control)
data:  treated and control
t = 3.0583, df = 15.485, p-value = 0.007736
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 0.03655926 0.20321347
sample estimates:
mean of x mean of y
0.1636364 0.0437500
```

	Bootstrap results	Classical stats
Mean difference	0.118	0.12
95% CI	0.05 – 0.19	0.04 – 0.20

Test for difference between 2 medians

```
a  = c(44, 51, 52, 55, 60, 62, 66, 68, 69, 71, 71, 76, 82, 91, 108)
b = c(52, 64, 68, 74, 79, 83, 84, 88, 95, 97, 101, 116)
n1 = length(a); n2 = length(b)
B = 1000
effect = numeric(B)
no.effect = 0
for (i in 1:B) {
   sampleA = sample(a, n1, replace=T)
   sampleB = sample(b, n2, replace=T)
   effect[i] = median(sampleA)-median(sampleB)
   if (effect[i] >= 0) no.effect=no.effect+1
  }

hist(effect, breaks=20)
quantile(effect, probs=c(0.025, 0.50, 0.975))
no.effect/1000
```




```
> quantile(effect, probs=c(0.025, 0.50, 0.975))
 2.5%   50% 97.5%
-31.5 -16.0  -2.5

> no.effect/1000
[1] 0.017
```

Bootstrap: summary

- Bootstrap approach: a very versatile way for solving many problems
- Based on concept of **resampling**
- Useful for statistical inference from “messy data”