
Non-parametric Computer Intensive Methods

Bootstrapping

Bootstrapping allows to approximate the sampling distribution of an estimate by *resampling*, that is, to randomly sample with replacement from the original sample. See Chapter 5.2 of Cohen's book.

- Procedure:

1. Resample the original sample with replacement, such that the size of the resample is equal to the size of the original sample.
2. Compute the statistic from the resample.
3. Repeat step 1 and 2 many times (1000 to 10000) to get a bootstrap sampling distribution of the statistic.

Bootstrapping

Relevant aspects of Bootstrapping:

- The original sample must be representative of the population.
- The sampling distribution obtained from bootstrap is centered at the observed statistic, not the population parameter. Therefore, the mean of the bootstrap sampling distribution of the means is not a good estimate of the mean of the population.
- Bootstrap is mainly used to estimate the standard deviation of the sampling distribution of the means, proportions, medians, etc.
- It works better with large sample sizes

Bootstrapping: Example 1

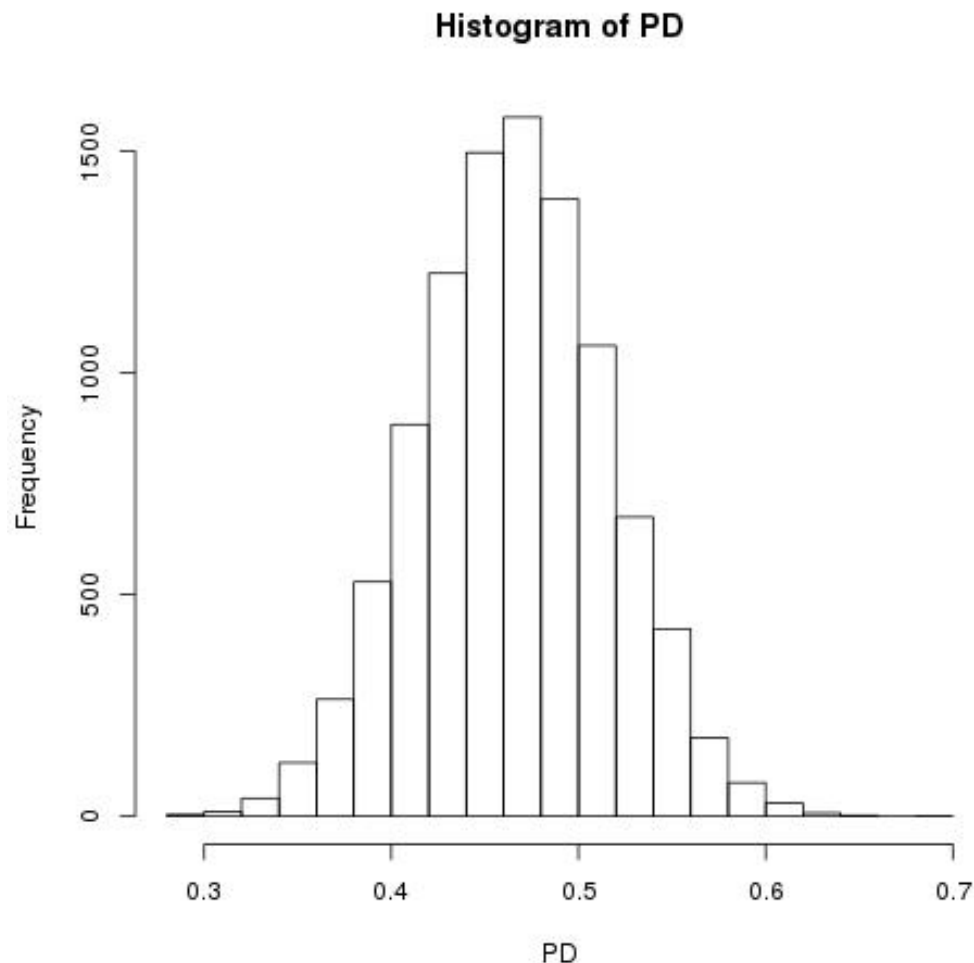
- Approximate the sampling distribution of proportions for the number of heads found in 100 coin flips.
- Generate a sample of size 100
- Randomly resample with replacement from the original sample and compute the proportion of heads for each sample.
- Resample several times to obtain a bootstrap sampling distribution of proportions.

Bootstrapping: Example 1

```
A = sample(c(0,1),100,replace=TRUE)
PD = c()
for (i in 1:10000){
  S = sample(A,replace=TRUE)
  p = sum(S)/100
  PD = c(PD,p)
}
hist(PD, freq=NULL,breaks=20)

# standard deviation of the bootstrap distribution
print(sd(PD))

# standard deviation of the sampling distribution
# of proportions for n=100
print(sqrt(0.5*0.5/100))
```



Bootstrapping: Example 2

One-sample test with censored data (Section 5.2.1 in Cohen's book)

- Random variable: Number of visited nodes of a randomized search algorithm to solve a problem, with a cut-off of 5000 nodes. The hypotheses are the following

$$H_0: \mu = 500$$

$$H_1: \mu < 500$$

- A sample of 10 runs, where 5000 means that the algorithm did not solve the problem after visiting 5000 nodes (censored value):

Trial	1	2	3	4	5	6	7	8	9	10
Data	287	610	545	400	123	5000	5000	601	483	250

How is the sample mean computed?

How to test the null hypothesis?

Bootstrapping: Example 2

Trial	1	2	3	4	5	6	7	8	9	10
Data	287	610	545	400	123	5000	5000	601	483	250

Mean of the 10 values: 1329.9

Censored mean of 8 values: 412.375

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} = 1.352$$

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} = -1.388$$

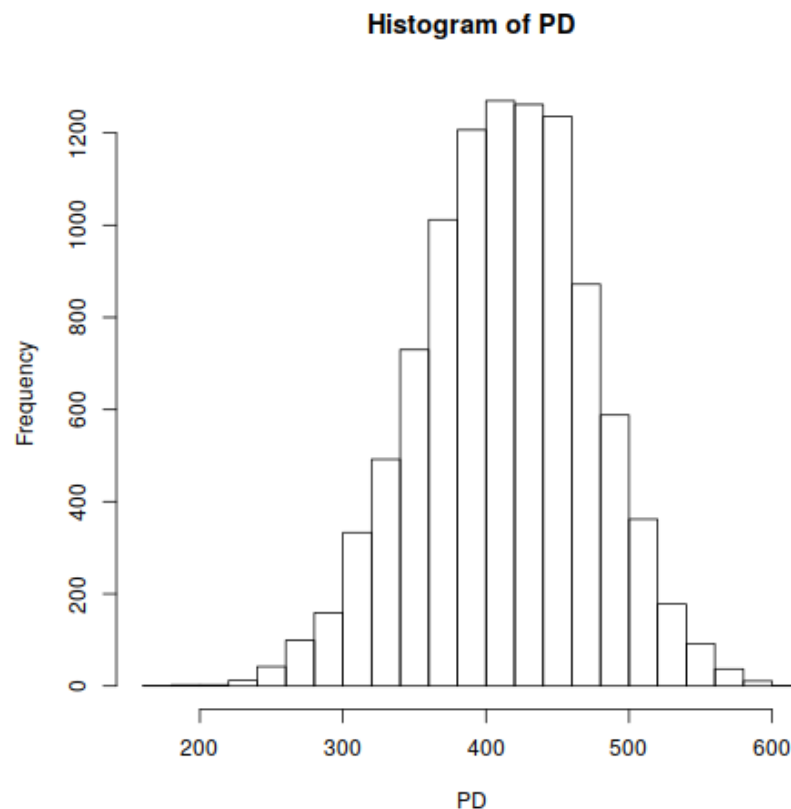
Which students' t distribution should be used as reference? $n = 10$ or $n = 8$?

Bootstrapping: Example 2

- Approximate the sampling distribution of the censored mean for $n = 10$ with bootstrap
- Randomly resample with replacement from the original sample for $n = 10$ and compute the censored mean. Due to the resampling procedure, several censored values may arise
- Repeat several times to obtain a bootstrap sampling distribution of the censored mean.

Bootstrapping: Example 2

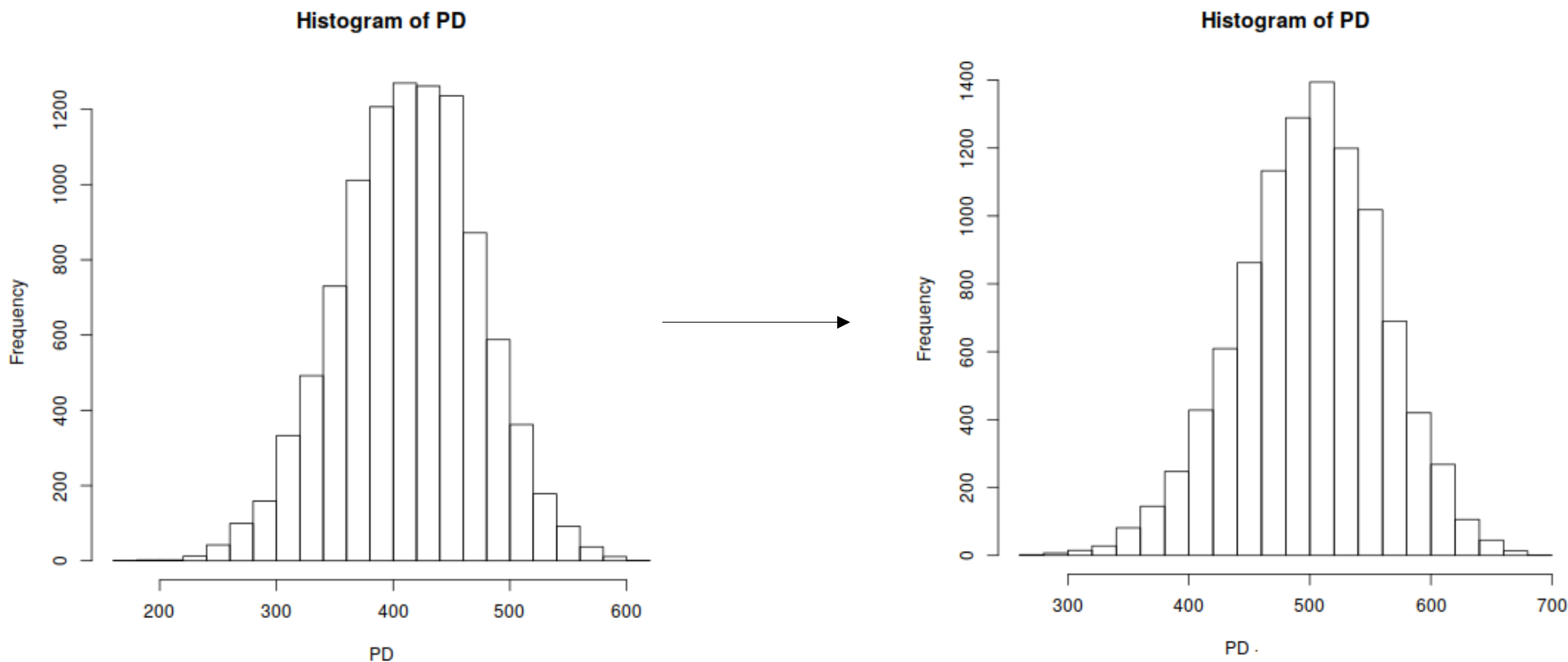
```
A = c(287, 610, 545, 400, 123, 5000, 5000, 601, 483, 250)
PD = c()
for (i in 1:10000){
  S = sample(A, replace=TRUE)
  p = mean(S[S<5000])
  PD = c(PD, p)
}
hist(PD, freq=NULL, breaks=20)
```



Bootstrap sampling distribution of the censored mean for $n=10$. The mean of the distribution is 412.75.

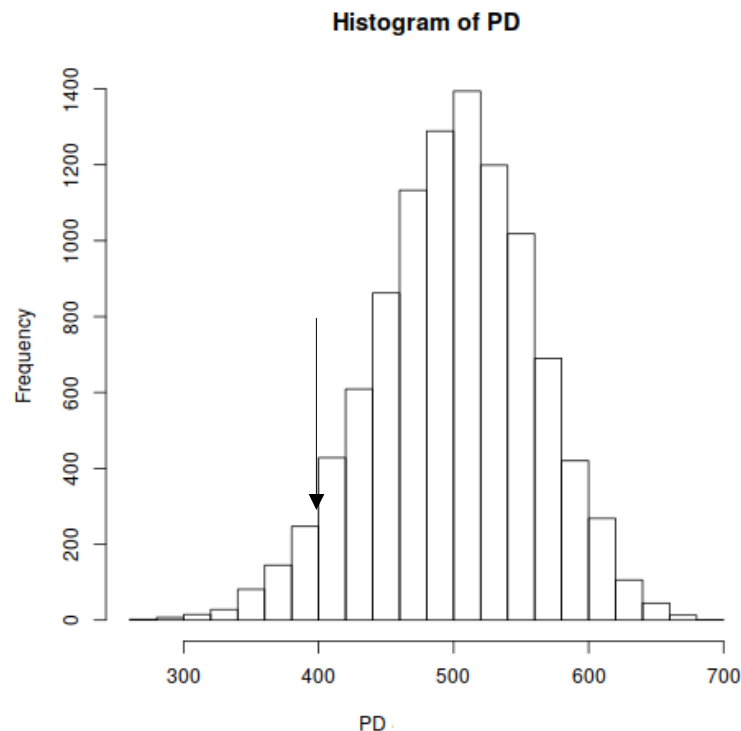
However, it cannot be used since the mean should be 500 under H_0

Bootstrapping: Example 2



The bootstrap sampling distribution has a similar shape to that of the sampling distribution, but with a different mean. Shift all values by the difference between 500 and the mean of the bootstrap distribution (412.75). The resulting distribution will be used as the sampling distribution under H_0 for the statistical test.

Bootstrapping: Example 2



Compute the critical value for 5% of significance level, which is the value v for which 5% of the sample means have a smaller value than v , which is 395.75. The censored mean in the original sample is 412.375. Therefore, H_0 is not rejected.

Randomization tests

- Statistical test in which the distribution of the test statistic under H_0 is obtained by calculating the values of the test statistic under several rearrangements of the elements in the original samples (see Section 5.3 of Cohen's book). It is mainly used as a two-sample statistical test.
- **Rationale:** Under the H_0 of no difference between the samples, any exchange of elements among the samples will produce no difference.
- **Procedure:** Similar to bootstrap but the resample is constructed without replacement.
- It does not perform any inference with respect to a population – it only tests whether there is an effect between the samples, by determining how likely they are under the rearrangements. But it has been shown to be very powerful.

Example - Hypothesis testing using randomization tests (two independent samples)

Assume you are the database administrator of a big information system. The database has just been installed and you are trying two tuning configurations: Conf. **A** and Conf. **B**. You expect Conf. B to be the fastest.

You use a given SQL package to test the execution time for each configuration.

After running several times the SQL package in both configurations you want to take a decision.

Question: Is configuration B faster than A?

Conf. A	Conf. B
----------------	----------------

74	69
----	----

66	71
----	----

88	80
----	----

68	88
----	----

79	64
----	----

68	65
----	----

87	74
----	----

79	76
----	----

78	89
----	----

72	68
----	----

86	67
----	----

85	72
----	----

86	
----	--

$$\mu_A = 78.15$$

$$\mu_B = 73.58$$

$$s_A = 7.94$$

$$s_B = 8.33$$

$$n = 13$$

$$n = 12$$

Example - Hypothesis testing using randomization tests (two independent samples)

- Record the difference between the means of the two original samples
- Resample without replacement, that is, randomly reassign the elements between the two groups and compute the difference between the means. Repeat several times in order to obtain an empirical distribution of the difference between the two sample means.
- Compute the *p-value* for a given significance value based on the empirical distribution

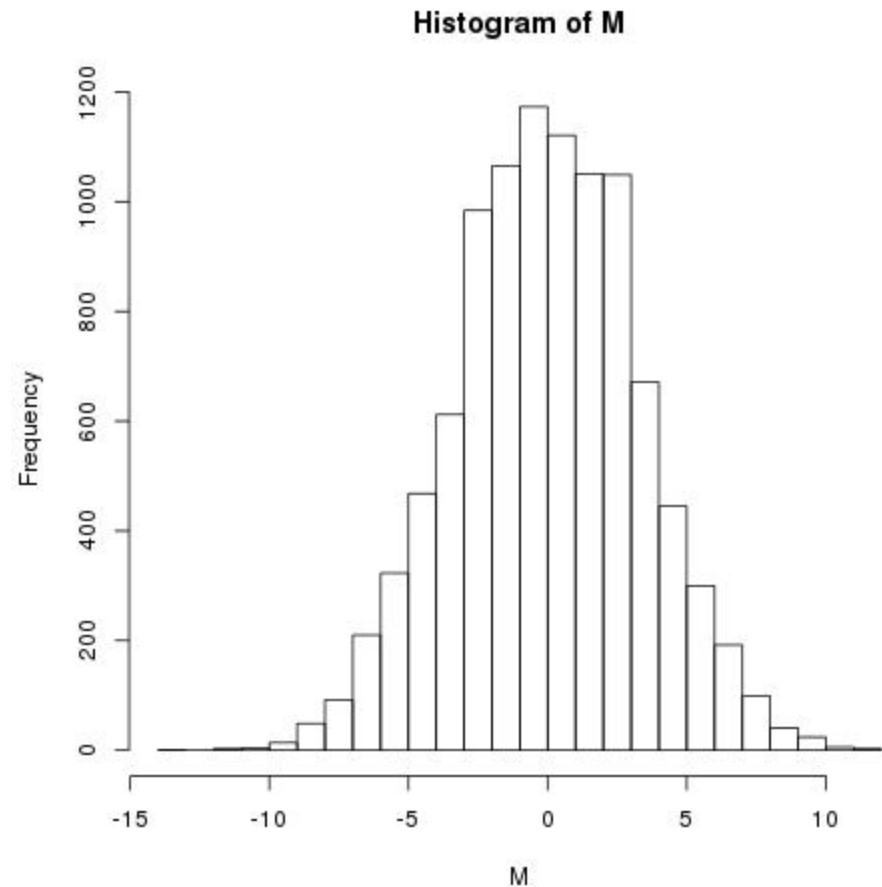
Example - Hypothesis testing using randomization tests (two independent samples)

```
A = c(74, 66, 88, 68, 79, 68, 87, 79, 78, 72, 86, 85, 86)
B = c(69, 71, 80, 88, 64, 65, 74, 76, 89, 68, 67, 72)

dif = mean(A) - mean(B)

C = c(A, B)
M = c()
pvalue = 0
for (i in 1:10000){
  P = sample(C, replace=FALSE)
  PA = P[1:length(A)]
  PB = P[(length(A)+1):length(P)]
  pdif = mean(PA) - mean(PB)
  M = c(M, pdif)
  if (pdif >= dif)
    pvalue = pvalue + 1
}

hist(M, breaks=20)
print(pvalue/10000)
```



P-value = 8.89% (it may change slightly if you run again)

Non-parametric alternative to Two-Way ANOVA

- Two-way ANOVA is known to work even when the assumptions of normality and homogeneity of variances are not met
- There are few non-parametric alternatives to two-way ANOVA with interactions, with known limitations:
 - ANOVA on ranks (perform ANOVA on the ranked values)
 - Randomization tests (compute F statistic for each permutation of the labels); see M.Anderson and C. Ter Braak, *Permutation tests for multi-factorial analysis of variance*, Journal of Statistical Computation and Simulation, 2003, Vol. 73 (2), pp. 85-113

Example: Two-Way ANOVA in R

Measurements	System A		System B		System C	
	Prog A	Prog B	Prog A	Prog B	Prog A	Prog B
1	0.0952	0.1432	0.1382	0.1082	0.0966	0.1066
2	0.0871	0.1343	0.1332	0.1032	0.1200	0.1100
3	0.0969	0.1314	0.1482	0.1182	0.1152	0.1252
4	0.1054	0.1443	0.1430	0.1130	0.1375	0.1275
5	0.0812	0.1312	0.1483	0.1083	0.1298	0.1398

Measurements of the time (in microseconds) required to perform a subroutine on three different systems programmed by two different programmers.

Is there a difference between systems ?

Is there a difference between programmers?

Is there any interaction between systems and programmers?

Example: Two-Way ANOVA in R

Test *interactions* and *main effects*

```
> aov.out = aov(Time~System*Programmer, data=D)
> summary(aov.out)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
System	2	0.000623	0.000312	2.965	0.0707 .
Programmer	1	0.000157	0.000157	1.493	0.2337
System:Programmer	2	0.007192	0.003596	34.219	9.38e-08 ***
Residuals	24	0.002522	0.000105		

Next: Compute F statistic for each random permutation of the observed times

Non-parametric Two-Way ANOVA

Randomization test with unrestricted permutations

(see Section 5.1 of M.Anderson and C. Ter Braak, 2003)

```
aov.out = aov(Time~System*Programmer, data=D)
FS = summary(aov.out)[[1]]$F[1]
FP = summary(aov.out)[[1]]$F[2]
FSP = summary(aov.out)[[1]]$F[3]
pvalueS = 0          # p-value for System
pvalueP = 0          # p-value for Programmer
pvalueSP = 0         # p-value for interaction
for (i in 1:5000){
  D$T = sample(D$Time)
  aov.out = aov(T~System*Programmer, data=D)
  pFS = summary(aov.out)[[1]]$F[1]
  pFP = summary(aov.out)[[1]]$F[2]
  pFSP = summary(aov.out)[[1]]$F[3]
  if (pFS >= FS)
    pvalueS = pvalueS + 1
  if (pFP >= FP)
    pvalueP = pvalueP + 1
  if (pFSP >= FSP)
    pvalueSP = pvalueSP + 1
}
print(pvalueS/5000)
print(pvalueP/5000)
print(pvalueSP/5000)
```

Output:

```
> 0.0712
> 0.222
> 0
```

**Interaction is significant at
significance level of 5%.**