Eltecon Data Science Course by Emarsys

Computational methods for measuring uncertainity

Tamás Koncz

October 21, 2020

Homeworks from last week

- Both Márton - Kamenár Gyöngyvér
- Emerson, Ian - Ralbovszki Judit
- Bat-Erdene, Boldmaa - Kashirin, Andrey

Section 1

Quick Recap

Why we do statistical inference?

- General goal: learn from (a limited) experience
- In statistical lingo: Observing a random sample, we wish to infer properties of the population it was drawn from
- In business: "If released, would our new product produce similar results than we observed in our experiment?"

Standard statistical methods

Calculate the 95% confidence interval as

$$\bar{x} \pm 1.96 * \frac{s}{\sqrt{n}}$$

where:

- $\bar{x}$ is the sample mean,
- s is the standard deviation of the sample distribution,
- n is the sample size

Drawbacks of standard statistical tests

- Parametric tests rely on certain assumptions, e.g. that sampling distribution is normal (which needs large n to be true)
- SE formula might not exists for other statistical estimators than the mean

Section 2

Bootstrapping

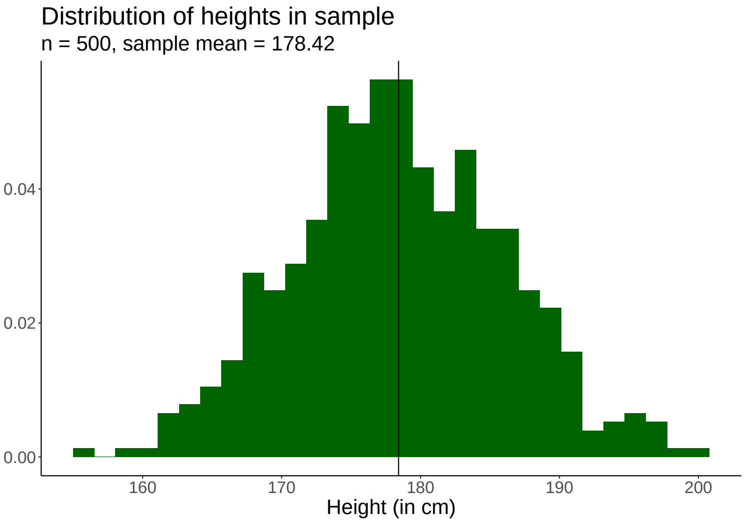
What is bootstrapping?

"The bootstrap is a data-based simulation method for statistical inference" - An Introduction to the Bootstrap

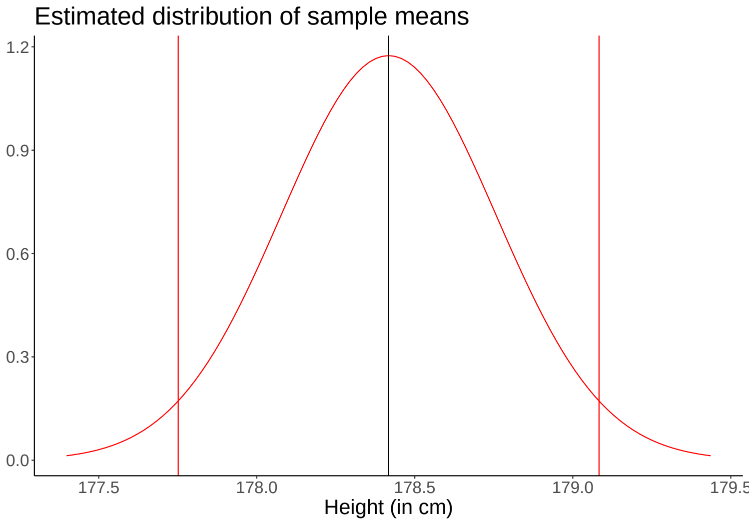
What is (non-parametric) bootstrapping?

- **data-based**
  - Gather a random sample from the population (assumed to be representative, e.g. iid)
- **simulation**
  - Re-sample with **replacement** to create another sample
  - One data point can appear 0, 1, or multiple times in a re-sample
  - Repeat this B times –> min. 10,000x
- **statistical inference**
  - Calculate the mean of each "new" sample
  - You can use the distribution of sample means to estimate the standard error, or to calculate confidence intervals

Example from last class



CI of sample means based on Student's t-distribution



Estimating the distribution of sample means with bootstrapping

```
height_sample[1:5]
```

```
## [1] 177.3506 187.9189 182.7978 186.8109 178.8722
```

Estimating the distribution of sample means with bootstrapping

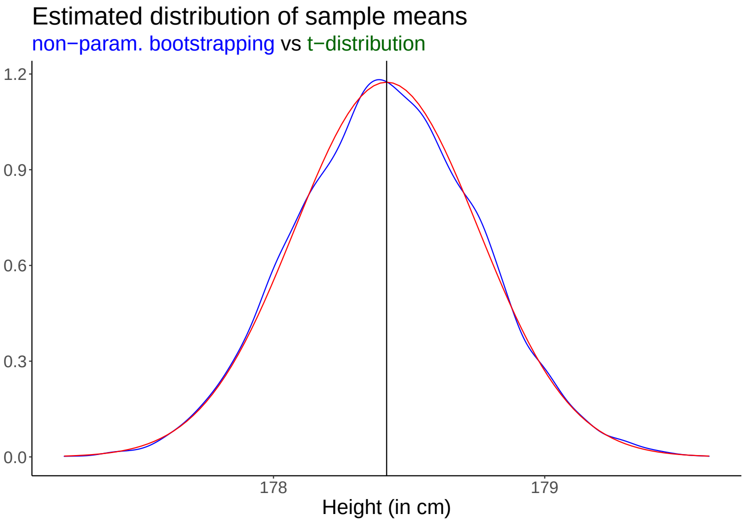
```
B = 10000
sample_size <- length(height_sample)
bs_sample_means <- data.table(
  sample_id = integer(), bs_sample_mean = numeric()
)

set.seed(1021)
for (i in 1:B) {
  bs_sample = sample(height_sample, sample_size, replace = TRUE)
  bs_sample_means <- rbind(
    bs_sample_means,
    data.table(sample_id = i, bs_sample_mean = mean(bs_sample))
  )
}
```

Estimating the distribution of sample means with bootstrapping

```r
head(bs_sample_means)
```

```
##    sample_id bs_sample_mean
## 1:         1       178.1933
## 2:         2       178.9084
## 3:         3       178.4169
## 4:         4       177.9684
## 5:         5       178.5280
## 6:         6       178.0378
```

Estimating the distribution of sample means with bootstrapping



Let's have a break!

Please be back in 15 minutes.

What is parametric bootstrapping?

- Based on your observed sample, you create a parametric model to fit the data
- With this model, you generate many new datasets
- Using these new datasets, you estimate the variation of your test statistic
- Not discussed any further in this class

Recap: Why bootstrap?

- You do not make any assumptions about how your test statistic is distributed. . .
- . . . while results should be very similar to what you get from statistical tests
- ("Fairly" recent development that we can do bootstrapping easily on our laptops)

Confidence intervals: the percentile method

lower bound:

```
bs_sample_means[, quantile(bs_sample_mean, 0.025, names = FALSE)]
```

```
## [1] 177.7636
```

```
sample_mean - 1.96 * (sample_sd / sqrt(sample_size))
```

```
## [1] 177.7514
```

upper bound:

```
bs_sample_means[, quantile(bs_sample_mean, 0.975, names = FALSE)]
```

```
## [1] 179.0843
```

```
sample_mean + 1.96 * (sample_sd / sqrt(sample_size))
```

Confidence intervals: the percentile method

(There are other methods, e.g. you could estimate the SE with bootstrap)

(We won't cover those in this class)

Confidence intervals: practice time

```
dt <- fread("experiment_result_HW.csv") %>%
    .[group == "treatment" & period == "first period"]

head(dt)
```

```
##    id       period     group has_viewed_website num_items_ordered sales_amount
## 1:  1 first period treatment                  0                 0            0
## 2:  9 first period treatment                  1                 2           15
## 3: 17 first period treatment                  0                 0            0
## 4: 32 first period treatment                  0                 0            0
## 5: 33 first period treatment                  0                 0            0
## 6: 40 first period treatment                  0                 0            0
```

Confidence intervals: practice time

```
dt[, .N]
```

```
## [1] 20058
```

```
click_rate <- dt[, mean(has_viewed_website)]
click_rate
```

```
## [1] 0.4548808
```

```
dt[, t.test(has_viewed_website)][["conf.int"]]
```

```
## [1] 0.4479890 0.4617727
## attr(,"conf.level")
## [1] 0.95
```

Confidence intervals: practice time

TODO:

- calculate the mean `click_rate` (= `mean(has_viewed_website)`)!
- calculate the median for `sales_amount` of people who ordered at least 5 items!

Confidence intervals: practice time

SOLUTION

Let's have a break!

Please be back in 15 minutes.

Hypothesis testing

Elements of hypothesis testing:

1. Specify H0
2. Define test statistic
3. Calculate distribution of test statistic under H0
4. Calculate p-value on from:

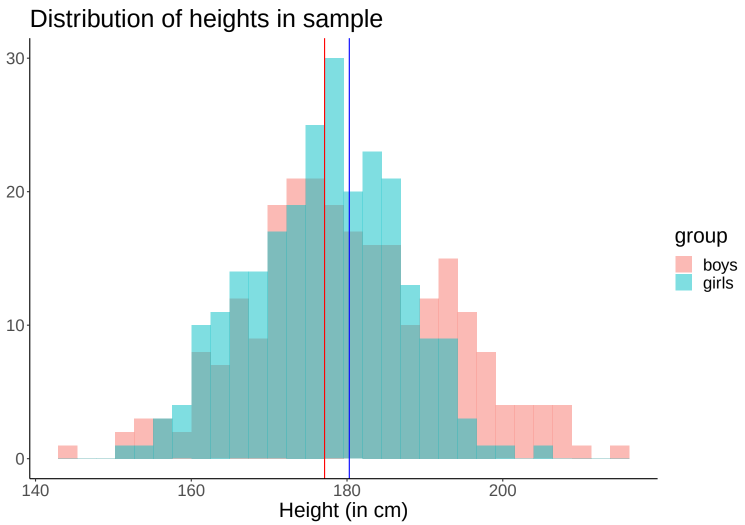
- test statistic on sample
- distribution of test statistic assuming H0

Hypothesis testing

Question: *"are girls the same height as boys on average?"*

H0: "The mean height of girls and boys are the same"

Hypothesis testing



Parametric hypothesis test

```r
height_sample[c(1, 2, 3, 498, 499, 500)]
```

```
##     group   height
## 1:  boys 177.4610
## 2:  boys 194.3703
## 3:  boys 186.1764
## 4: girls 187.0898
## 5: girls 176.2139
## 6: girls 182.5784
```

Parametric hypothesis test

```
t.test(
  height_sample[`group` == "boys", height],
  height_sample[group == "girls", height]
)[["p.value"]]
```

```
## [1] 0.001695985
```

Permutation hypothesis testing

Define test statistic:

```
sample_height_diff <- abs(sample_mean_height_girls - sample_mean_height_boys)
sample_height_diff
```

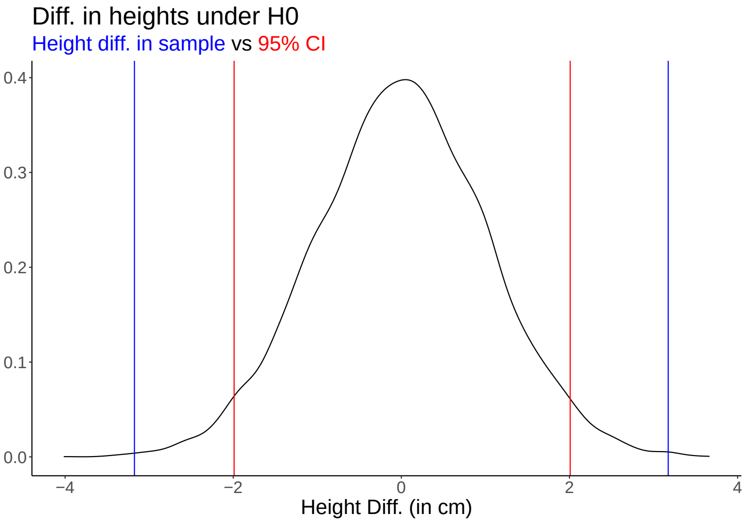
```
## [1] 3.175253
```

Permutation hypothesis testing

Calculate distribution of test statistic under H0

```
B = 10000
perm_sample_diffs <- data.table(perm_id = integer(), perm_diff = numeric())

set.seed(1021)
for (i in 1:B) {
  perm_sample <- data.table(
    group = height_sample[, group],
    height = sample(height_sample[, height], sample_size, replace = FALSE)
  )

  perm_sample_diffs <- rbind(
    perm_sample_diffs,
    data.table(
      perm_id = i,
      perm_diff = perm_sample[group == "boys", mean(height)] - perm_sample[group == "girls", mean(height)]
    )
  )
}
```

Permutation hypothesis testing



Permutation hypothesis testing

Calculate p-value

```
p_value <- perm_sample_diffs[,
  sum(sample_height_diff < abs(perm_diff)) / .N
]

print(p_value)
```

```
## [1] 0.0021
```

Why use the permutation approach?

- Works better on small samples
- Relies on no assumptions (compared to parametric approaches)
- Comparing more special test statistics

Hypothesis testing - practice time

```r
dt <- fread("experiment_result_HW.csv") %>%
    .[period == "first period", .(group, sales_amount)]

head(dt)
```

```
##        group sales_amount
## 1: treatment            0
## 2:   control            6
## 3:   control            0
## 4:   control            6
## 5:   control            0
## 6:   control            0
```

Hypothesis testing - practice time

TODOs:

- Plot distribution of mean diff. in `Sales Amount` (using bootstrapping)
- Calculate CIs for Treatment vs Control avg. `Sales Amount` (using bootstrapping)
- Calculate p-value for H0: Treatment and Control `Sales Amount`-s are the same! Calculate using `t.test()` and with a permutation test as well.
- Use `seed = 1021` for randomization!
- Use `B = 10000`!

Hypothesis testing - practice time

SOLUTION

The {boot} package

```r
library(boot)
boot::boot() # Bootstrap Resampling
boot::boot.ci() # Nonparametric Bootstrap Confidence Intervals
```

Drawbacks of bootstrapping

- The naive bootstrap (discussed here) is built on **large sample theory**, hence needs a sizeable sample to work well
- Not suitable for estimating extreme values (e.g. 99th percentile)
- Won't increase the number of information in your data!
- Depends on your sample being an unbiased representation of the population

Section 3

Homework

Homework

- Task:
  - Use `experiment_result_HW.csv` or your own project's data
  - Calculate the point estimates and add uncertainty with bootstrapping to one of your KPIs
  - Calculate p-value with the permutation method for the difference of Treatment / Control groups
- Deadline: next class (2020-11-04)
- Presenters:
  - Im Seongwon - Kim Yeonggyeong
  - Szőnyi Máté - Tran, Dung
  - Sármány Áron - Schmall Róbert