Eltecon Data Science Course by Emarsys

Simulating the uncertainty of measurement

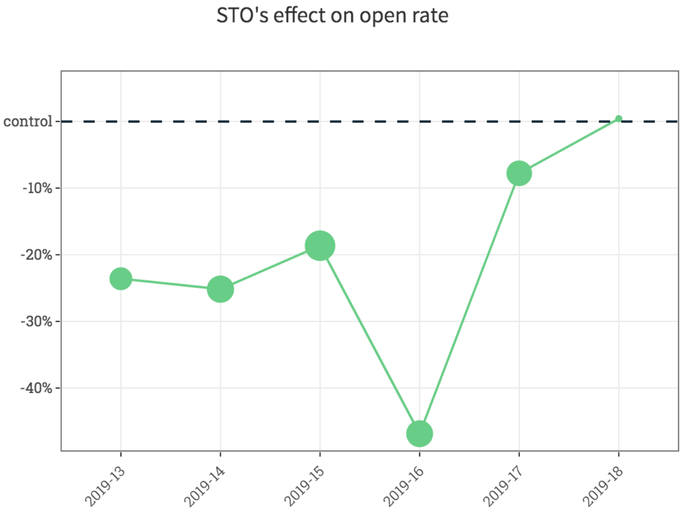
András Bérczi

October 16, 2019

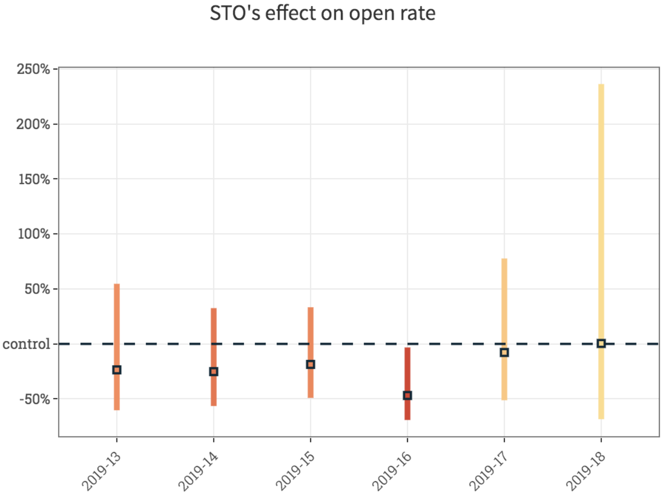
There is always an effect. . .

- We can always measure something.
- Is there really an effect?

There is always an effect. . .



But not necessarily significant!



Why do have uncertainty in the measurement?

- If you knew the whole population, there wouldn't be uncertainty in your measurement
- But we only see 1 'segment' of the data = we have a sample of the population

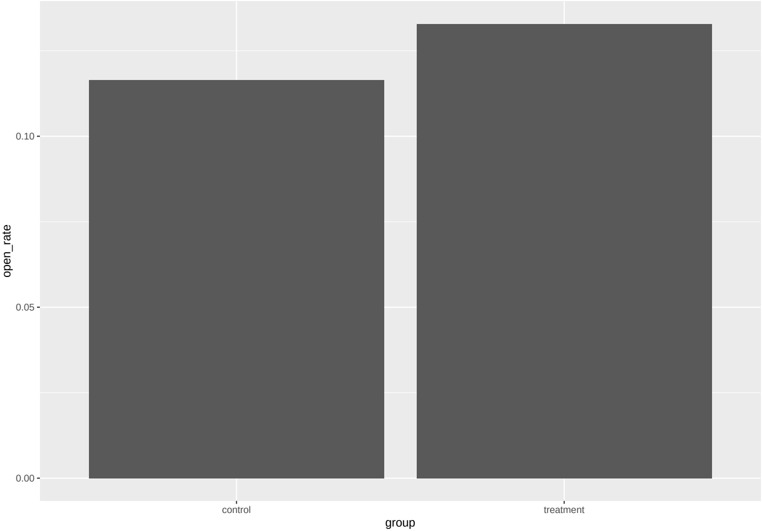
How can we calculate uncertainty to our measurement?

- We know the distribution −> calculate variance
- Monte-Carlo method
- Bootstrapping
- Permutation test

Calculate uncertainty for an experiment

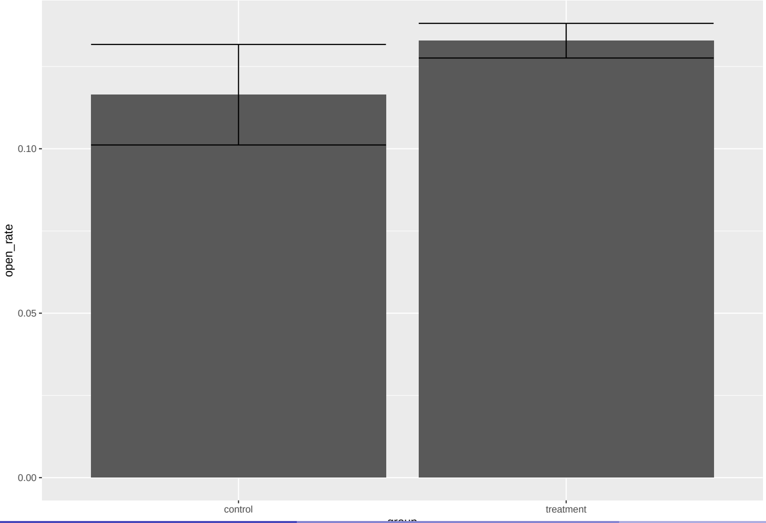
contact_id	group	num_send	num_open	num_click	sales_amou
1	treatment	0	0	0	
2	treatment	3	0	0	
3	treatment	2	1	0	
4	treatment	3	0	0	
5	treatment	0	0	0	
6	treatment	0	0	0	

Results from an experiment:



Are the results significant? Calculate the variance!

- Assumption about the distribution of the data



Now your turn!

1. Calculate the click rate and the uncertainty!

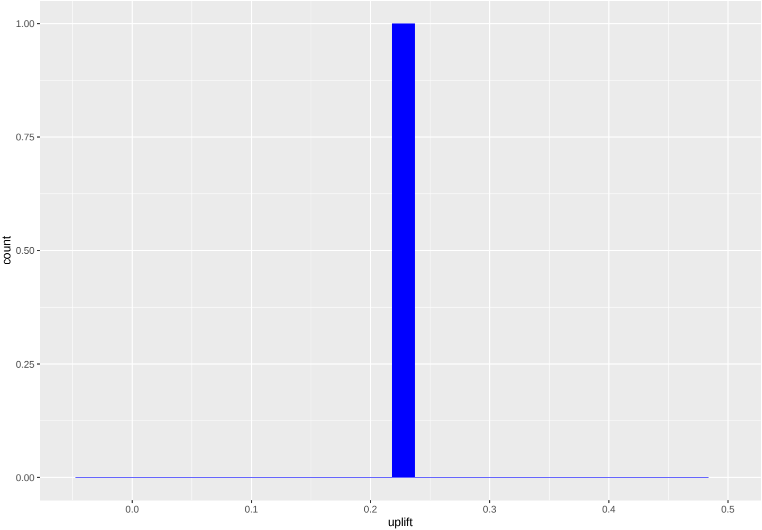
2. Plot the results! What do you see on the plots? Are the results significant?

Monte-Carlo method

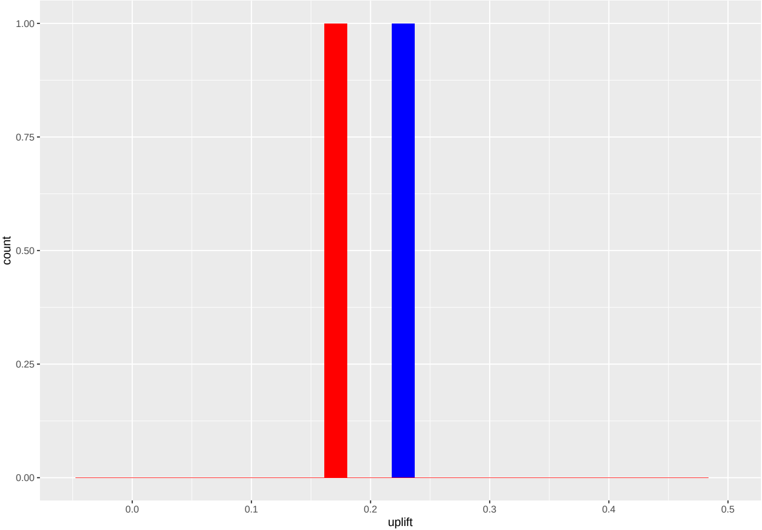
- Pick repeatedly from a distribution(s)
- Use randomness to show uncertainty
- Useful, when you do not have a closed form to calculate the variance
- We still need to know the distribution of our variable(s)!

How to calculate uncertainty with Monte-Carlo method

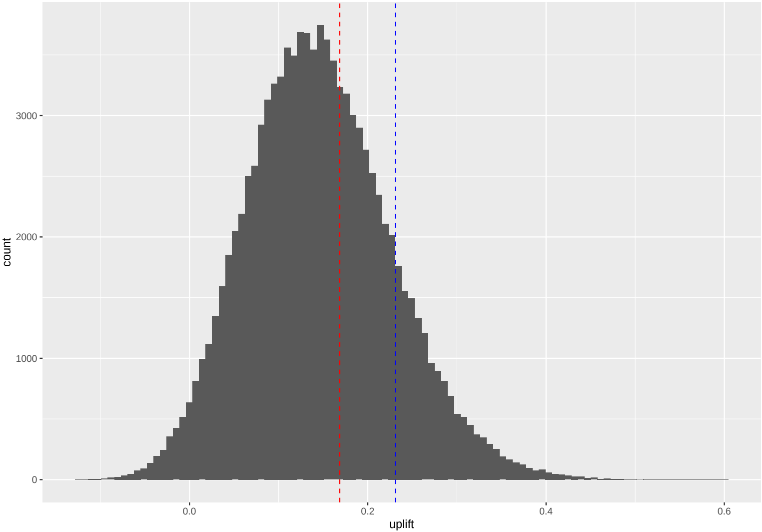
Draw samples from the sampling distribution of the mean from both groups and calculate the uplift



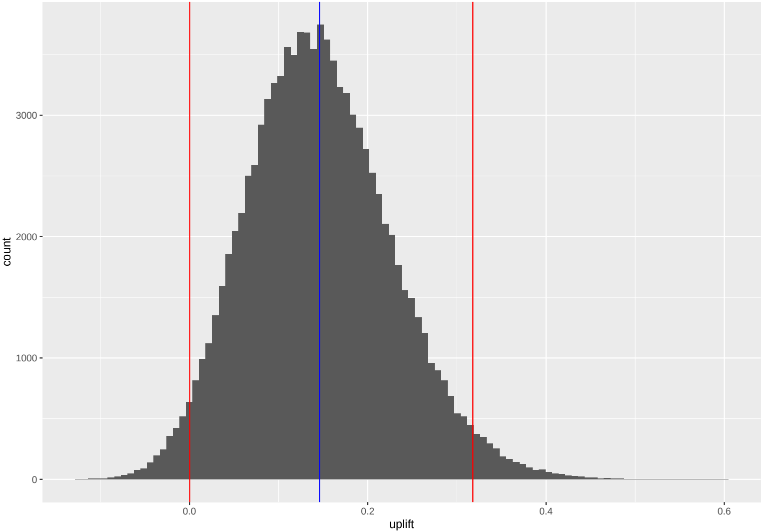
Do it again!



Repeat it N (let's say 100 000) times!



Calculate the mean and the confidence intervals! Is our treatment effective based on the open rate?



Your turn!

3. Calculate uncertainty of effect on the click rate with Monte-Carlo method

4. Plot and interpret the results!

How would we do the same if we do not know the distribution?

Bootstrapping

- resampling with replacement
- quantify the uncertainty associated with a given estimator
- computationally heavy calculation

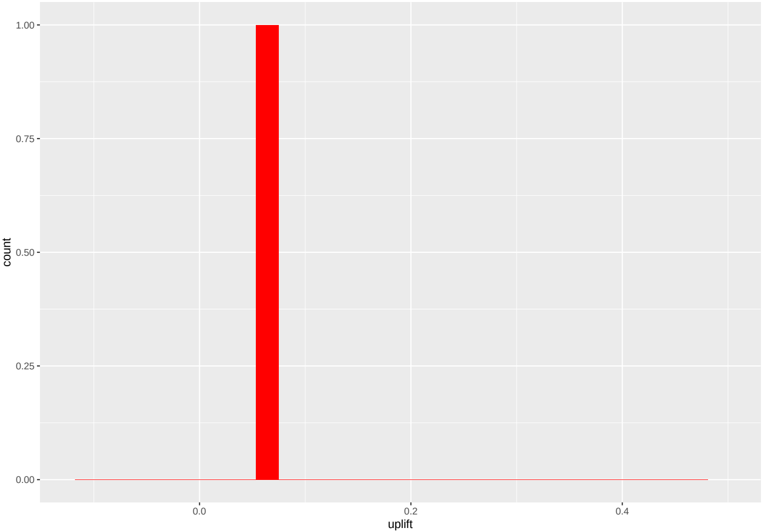
How bootstrapping works

Sample with replacement from original data

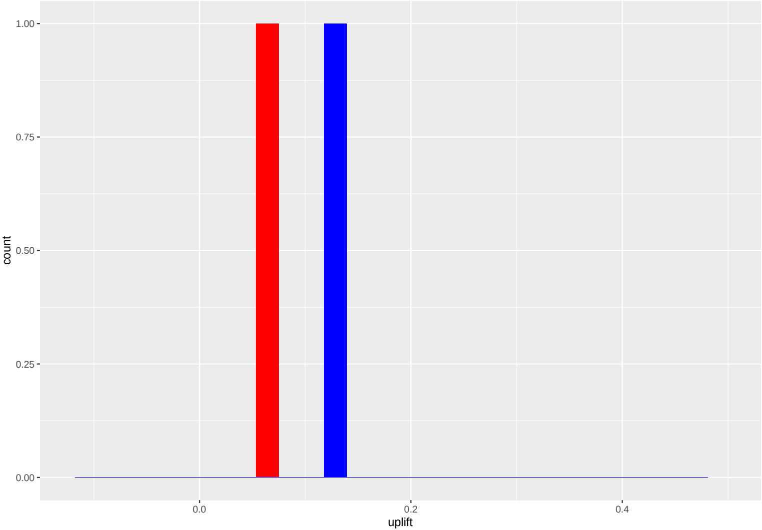
```
dt[sample(.N, .N, replace = TRUE)]
```

```
##        contact_id     group num_send num_open num_click sal
##     1:         58 treatment        1        0         0
##     2:       3185 treatment        0        0         0
##     3:       6861 treatment        3        0         0
##     4:       7418 treatment        0        0         0
##     5:       8835 treatment        3        1         0
##   ---
##  9996:       3001   control        3        0         0
##  9997:       7651 treatment        3        0         0
##  9998:        869   control        0        0         0
##  9999:       4622 treatment        3        0         0
## 10000:       7025 treatment        3        0         0
```

Calculate your statistic for bootstrap sample



Create another bootstrap sample and calculate statistic



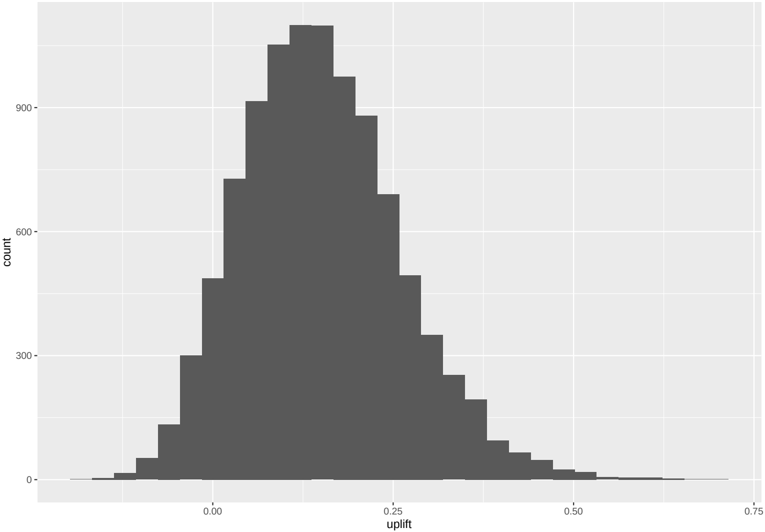
Repeat it N times (let's say N=1000)...

```
set.seed(1234)
bootstrapped_stats <- map_df(1:10000, ~{
    dt[sample(.N, .N, replace = TRUE)] %>%
        .[,
            .(bootstrap_id = .x,
              open_rate = sum(num_open) / sum(num_send),
              num_send = sum(num_send)),
            by = group
        ]
})
```

. . . so we could get a distribution of uplifts

```
##        bootstrap_id    control treatment      uplift
##     1:            1 0.09751773 0.1302057  0.33520015
##     2:            2 0.11036174 0.1345331  0.21901935
##     3:            3 0.12220917 0.1276690  0.04467623
##     4:            4 0.11049107 0.1307301  0.18317299
##     5:            5 0.12875289 0.1270687 -0.01308072
##    ---
##  9996:         9996 0.11649295 0.1301486  0.11722281
##  9997:         9997 0.11498856 0.1350342  0.17432749
##  9998:         9998 0.11170848 0.1372134  0.22831639
##  9999:         9999 0.14251497 0.1314651 -0.07753472
## 10000:        10000 0.10998811 0.1336017  0.21469231
```

Distribution of uplifts

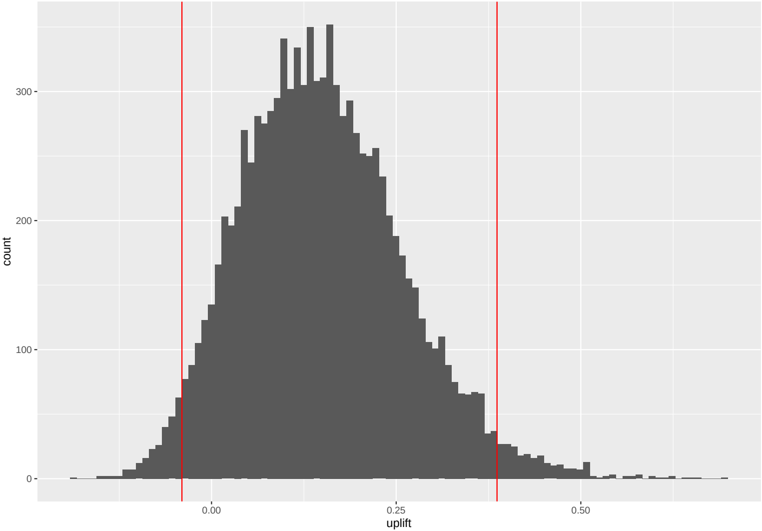


Calculate confidence intervals from distribution

```
CI_from_bs <- bs_uplift[, .(
    CI_lower = quantile(uplift, 0.025),
    CI_higher = quantile(uplift, 0.975)
)]
CI_from_bs
```

```
##       CI_lower CI_higher
## 1: -0.04010245 0.3865375
```

Calculate confidence intervals from distribution



Your turn!

5. Calculate the uncertainty of effect with bootstrapping for 'sales amount per contact'

6. Plot the results!