# Summary and resampling statistics

Spyros Samothrakis
Research Fellow, IADS
University of Essex

January 24, 2017

# Summary statistics and resampling statistics

- Today we are going to discuss summary statistics and resampling statistics
  - Summary statistics try to capture the "essence" of a set of observations
  - Resampling statistics try to find out how if we can "trust" a parameter we inferred from the observations
    - If we re-did the sampling, would we still found out the same thing?
- Resampling statistics are far more intuitive to understand then using t-tests (I think...)

# An example problem

- Let's say that a journalist was tasked with finding the salaries of a bigger corporation
- But could only find through friends and acquaintances the salaries of certain employees

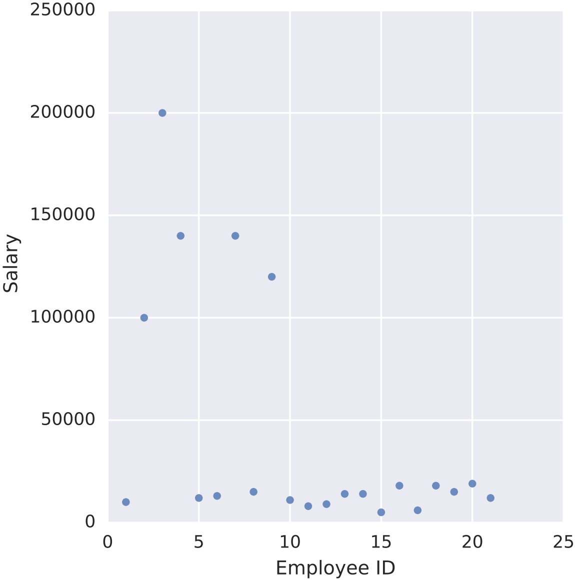
| Employee ID | Salary |
|---|---|
| 1 | 10000 |
| 2 | 100000 |
| 3 | 200000 |
| 4 | 140000 |
| 5 | 12000 |
| 6 | 13000 |
| 7 | 140000 |
| 8 | 15000 |
| 9 | 120000 |

# (Continued table)

| Employee ID | Salary |
|---|---|
| 10 | 11000 |
| 11 | 8000 |
| 12 | 9000 |
| 13 | 14000 |
| 14 | 14000 |
| 15 | 5000 |
| 16 | 18000 |
| 17 | 6000 |
| 18 | 18000 |
| 19 | 15000 |
| 20 | 19000 |
| 21 | 12000 |

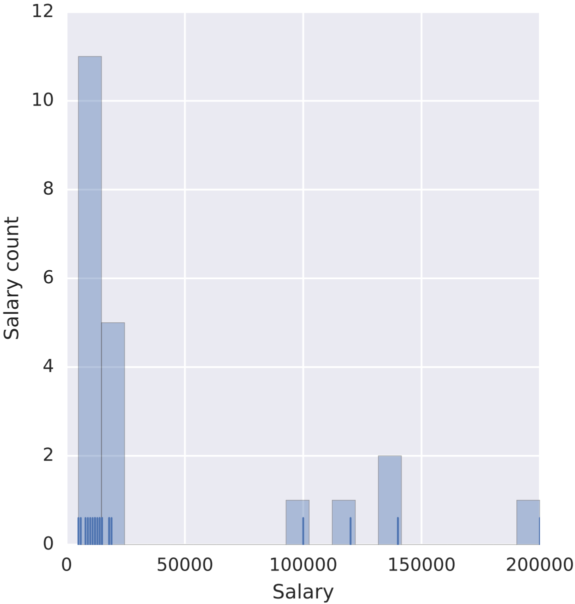
# Let's plot

```
df = pd.read_csv('./customers.csv')

# There are far
# better ways of doing this
data = df.values.T[1]

sns_plot = sns.distplot(df,
bins=20,
kde=False,
rug=True).get_figure()
```



# Histogram plot

```
sns_plot2 = sns.distplot(data,
bins=20,
kde=False,
rug=True).get_figure()
```



# Measures of central tendency

- (Sample) mean
  - $\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$
- (Sample) median
  - Rank $X_i$
  - $M = \begin{cases} X_{N/2+1} & \text{if } n \textit{ is odd} \\ (X_{N/2} + X_{(N+1)/2})/2 & \text{if } n \textit{ is even} \end{cases}$
- In the salary data
  - $\mu = 42809.523810$
  - $M = 14000.000000$

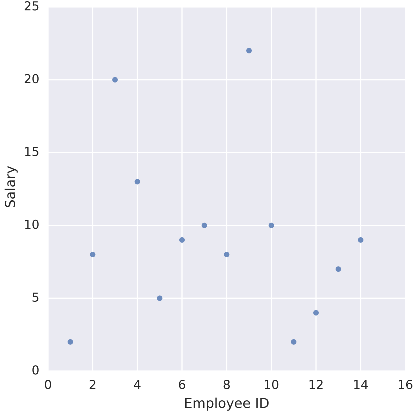
# Measurements of dispersion

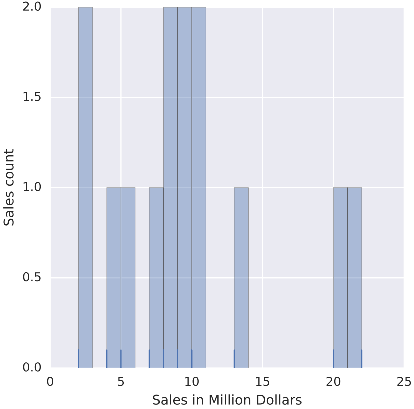
- (Sample) Standard deviation
  - $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$
  - Variance is $\sigma^2$
- Median absolute deviation
  - $MAD = M(|X_i - M(X)|)$
- In our data we have:
  - $\sigma^2 = 3230916099.773242$
  - $\sigma = 56841.147946$
  - $MAD : 4000.000000$

# Sales

- A company has recorded their sales for 14 days
- They want to understand their data
- Let's plot

# Histogram plot of Sales



# Summary statistics

$\mu = 9.214$

$M : 8.500000$

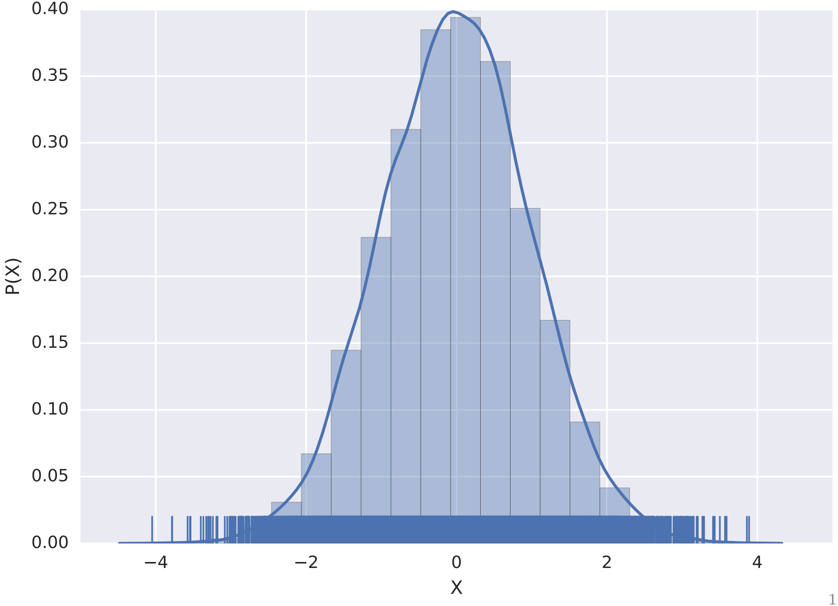
$\sigma^2 : 32.311$

$\sigma : 5.684296$

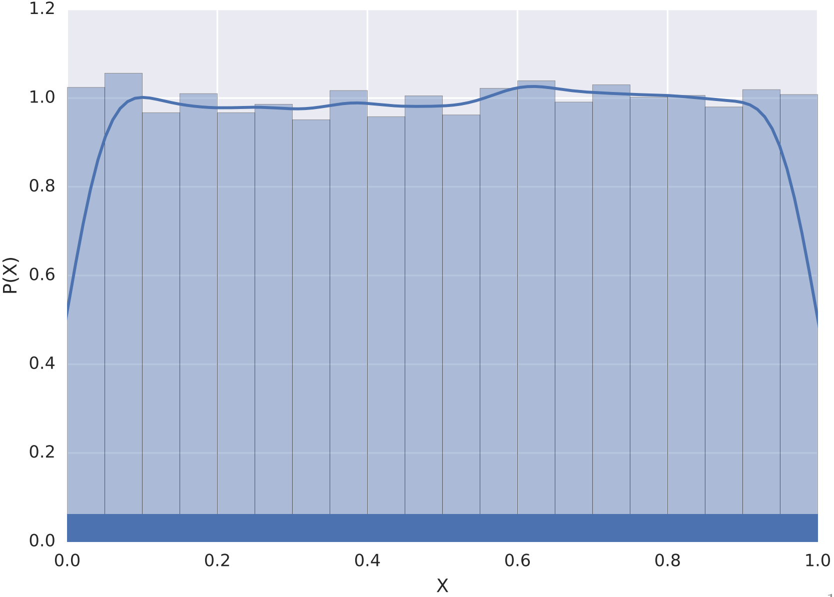
$M : 2.500$

Note that there are tons of other summary statistics, this is practically for illustration purposes only

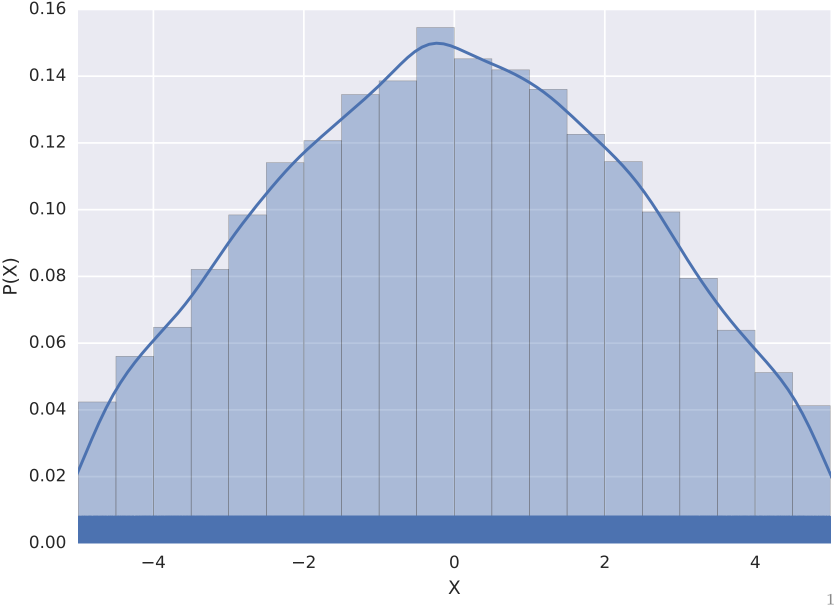
# Normal distribution



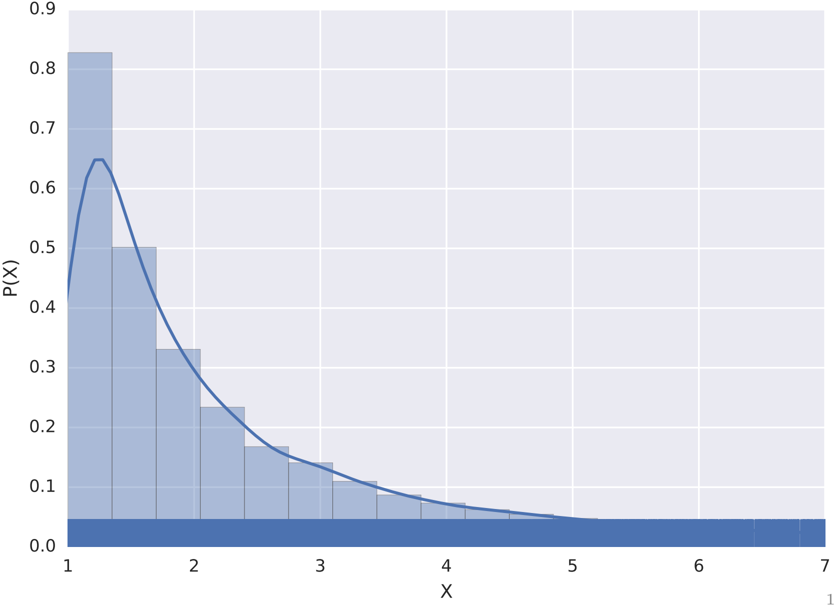
# Uniform distribution



# Normal high variance distribution



# Pareto distribution



# Are we confident we got the right mean?

- How confident should the journalist or the analyst be about their summary statistics?
- If they sampled another 14 days, maybe the sale numbers would be completely different?
- We if possible we would like to build something termed "confidence intervals" (CI)
  - Get a measure of "If I do this sampling process a lot of times, how many of them would I actually see a certain range"
- We are going to take the above statement seriously
  - And introduce the bootstrap!

# The bootstrap

- We are going to use a method called the bootstrap to create those CIs
- Very popular, computational method
- DiCiccio, Thomas J., and Bradley Efron. "Bootstrap confidence intervals." Statistical science (1996): 189-212.
- You will see this name (bootstrap) used quite often in scientific contexts
  - It refers to a self-starting process
  - The mind "understanding itself"
  - Pulling yourself up by the bootstraps
- Hard to do without a machine

# Bootstrapping (1)

- Ideally, so that we can find our confidence interval we would sample from the population
  - i.e. the journalist would go over to a different set of friends (possibly interloping)
  - Ask them to get her some salaries
  - Repeat
- Once we have a collection of different means we can say that a mean will fall within a certain range with a certain probability
  - But this is almost impossible
- We can use our sample however in a smart way
  - Resample from the sample!

# Bootstrapping (2)

- Sample with replacement from the data you have already
  - Create $\{1 ... B\}$ bootstraps of the same size
  - Let's assume each observation in the initial dataset is $X_i$, where $i$ is the order appearing

$$X^1 = X_4^1, X_5^1, X_3^1, X_5^1 ...$$

$$X^2 = X_3^2, X_7^2, X_7^2, X_8^2 ...$$

$$X^{\cdots} = ...$$

$$X^B = X_8^B, X_3^B, X_2^4, X_4^1 ...$$

# Bootstrapping (3)

- Let's do one one example
- $X = \{1,0,1,2\}$
- Let's draw three samples
  - I will simulate the dice rolls

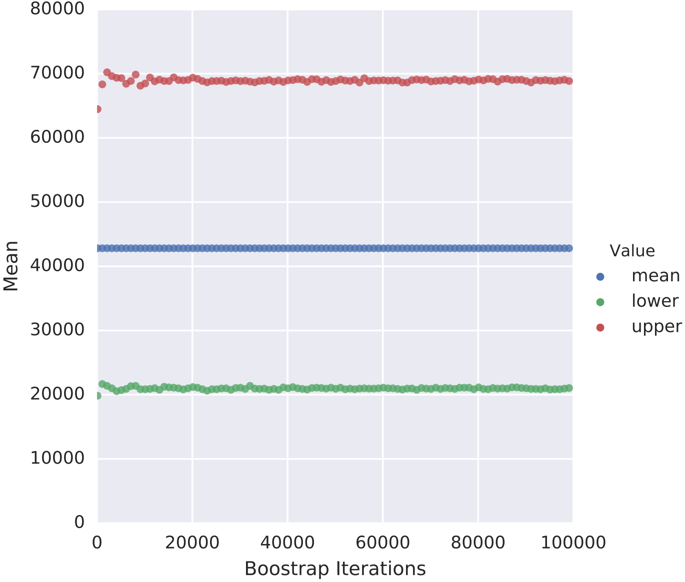
# Bootstrapping (4)

- Get the mean for each sample (since this is what we are interested in)
- We can now rank the means
- We remove the bottom 10% and the top 10% to find $\gamma = 0.80$
- For the salary data

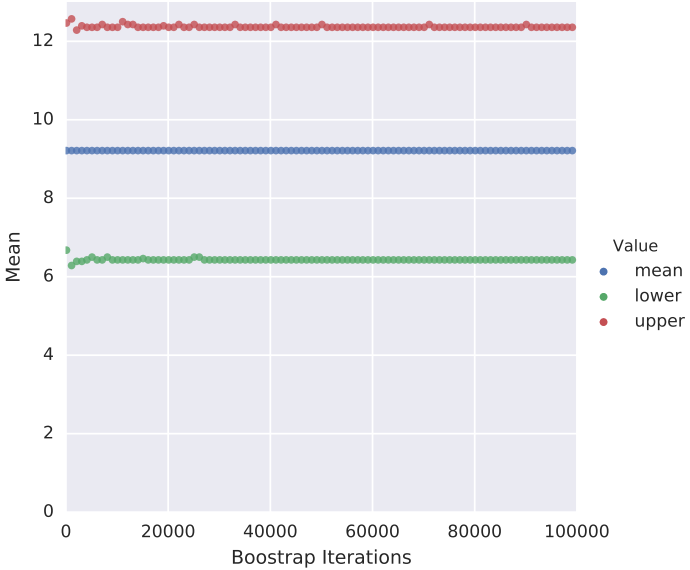
$$X = [6.86, 7.29, 7.86, 8.14$$
$$8.36, 8.79, 8.86, 9.14$$
$$9.29, 9.5, 9.5, 9.71$$
$$10.36, 11.14, 11.14, 13.21]$$

- What about if I was interested in $\gamma = 0.90$?
- What about if I was interested in $\gamma = 0.95$?

# Salaries


80000
70000
60000
50000
40000
30000
20000
10000
0
Mean
0
20000
40000
60000
80000
100000
Boostrap Iterations
Value
mean
lower
upper

# Sales


12
10
8
6
4
2
0
Mean
0
20000
40000
60000
80000
100000
Boostrap Iterations
Value
mean
lower
upper

# What can we say about the means now?

- Salaries mean is
- Sales mean is
- We can do bootstrap to estimate *any* quantity we want as long as the distribution has a defined variance and mean
  - i.e. not always
- But for most practical matters, yes
- No need for any other test

# Data bias

- I have described a very biased process of collecting samples above
  - The journalist asked her friends
  - All her friends love football
  - What he might actually have learned is the salary of football loving employees
- How about the sales?
  - Was there anything extra-ordinary on the day these measurements where taken?
  - Maybe it was Christmas
- Be very careful to randomise properly, and if not at least take care to state your bias

# A/B Testing

- Suppose you had two versions of a website
  - and you would like to check if the newer version is better
- Two version of an e-mail
  - and you would like to check if the newer, fancier version is better
- A new drug
  - and you would like to see if it actually cures
- A zombie apocalypse
  - and you have found a serum to cure zombiness

# Hypothesis testing

- Same as A/B testing
- The name people used to call the same procedure when testing for
  - Drug effects
  - Physical effects
  - Quality management
- A lot of Data science concepts are just “re-imigani

# Example problem

- A company sends out e-mails
  - Various promotions and news content
  - They want users to click on the links and get on their website
  - They already have an e-mail format
  - Mark from marketing comes up with an e-mail with improved content
- Is it better?
  - Without causing too much disruption

# Hypothesis testing

- They send 11 e-mails of of the usual type (control)
- They also send 11 e-mails of the new design (test)

```
old = np.array([0,1,1,1,0,1,1,0,0,1,0])
new = np.array([0,1,1,0,1,1,0,1,1,1,0])
```

$\mu_{old} = 0.18$

$\mu_{new} = 0.455$

$t_{obs} = \mu_{new} - \mu_{old} = 0.27$

Should they change?

# Hypothesis forming

$H_0$: The two e-mails make no difference (their means are equal) - this is called the *null* hypothesis

$H_1$: The second e-mail is better,and thus has a higher mean

- Set $\alpha = 0.05$, or equivalently the 95% CI $t_{obs}$ does not contain $H_1$
- What is the probability of observing something as extreme as we just observed by pure chance?

# Permutation testing (1)

- Merge all the data into a new array

```
array([0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0])
```

- Permute it random, i.e. form a new array from the same elements

```
array([0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0])
```

# Permutation testing (2)

- Split again into new and old

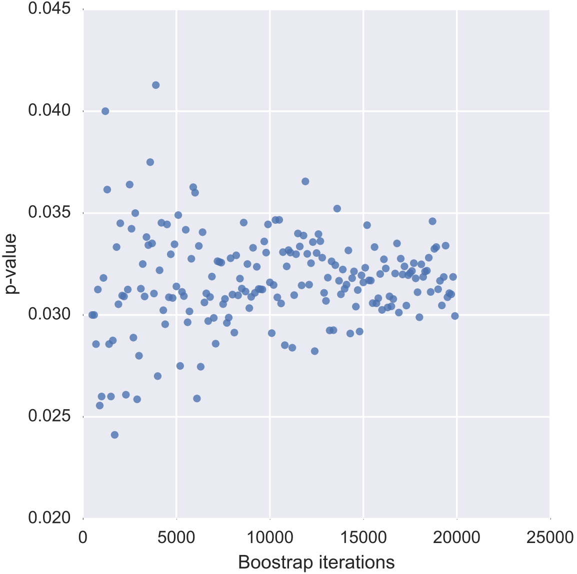
```
pold = np.array([0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1])
pnew = np.array([0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0])
```

- Record if the value of the test was more extreme or not
  - $t_{perm} = \mu_{pnew} - \mu_{pold}$
  - $t_{perm} > t_{obs}$
- Keep on permuting and recording
- Find the number of times $t_{perm} > t_{obs}$
  - Divide by the number of permutations you used
- You call that number your $p - value$

# Permutation tests (3)

- If you do this for 19,000 permutations you get $p - value = 0.032$
- Hence we can conclude 5 out of a 100 times you will get a higher difference in means
- Find out if this number is smaller than than $\alpha = 0.05$
- If yes, you can reject the $H_0$ (which it is)

# Permutation test (4)



# Another experiment

- Bob decides that adding a sound to the e-mail should increase user clicking even more
- Thinking that it his solution is better for sure, he sends more e-mails with sounds (i.e. the new version)
  - Not exactly A/B testing, but he seems eager...
- Results come back and he had to somehow show that his new e-mail procedure is better

# Some data analysis

```
old = np.array([0,1,1,1,0,1,1,0,0,1,0])
new = np.array([0,1,1,0,1,1,0,1,1,1,0,0,1,1,1,1,1,1,1])
```

$\mu_{old} = 0.546$

$\mu_{new} = 0.73$

$t_{obs} = \mu_{new} - \mu_{old} = 0.19$

# Results

- With $19,000$ permutations we get a $p = 0.07$
- Thus we have failed to reject the null hypothesis
- Does not mean that the sound doesn't have any impact
- Just that we can't tell the impact

# Errors

- Type I error: Rejecting $H_0$ even though it is true
- Type II error: Failing to reject $H_0$ even though it is false

| | $H_0$ is true | $H_0$ is false |
|---|---|---|
| Reject $H_0$ | Type I error (false positive) | Correct Inference |
| Fail to reject $H_0$ | Correct inference | Type II error (false negative) |

# Specificity and Sensitivity

- Specificity refers to the level we set $\alpha$
  - $\alpha$ is "false positive rate"
  - The higher, the more susceptible the test is to Type I errors
  - Think of this as raising false alarms
- Sensitivity refers to another parameter, which we haven't set at all for now, called $\beta$
  - $\beta$ is "false negative rate"
  - The higher it is, the more we are bound to do Type II errors
  - Think of this as failure to detect a phenomenon
- "Surely you only need one of them!" (No!)

# Power analysis

- A question that would naturally rise up is how many samples do we need to collect if we are to perform a study within a certain error
- No easy solution
- In practice, sample as much as you can
- See previous studies in the literature
- If you have done a study before, use the boostrap!
  - How?
- You might be tempted to increase $\alpha$, but this will increase your chance for a Type I error

# A more "hackish idea"

- Get the confidence intervals for both populations
- If they overlap, fail to reject $H_0$
- If not, reject $H_0$
- Very tempting to do this
  - Actually you can
  - It's a bit more conservative, but people do it all the time
  - Not thaaaaat bad if the samples are independent

Schenker, Nathaniel, and Jane F. Gentleman. "On judging the significance of differences by examining the overlap between confidence intervals." The American Statistician 55.3 (2001): 182-186.

# P-Hacking

"In the course of collecting and analyzing data, researchers have many decisions to make: Should more data be collected? Should some observations be excluded? Which conditions should be combined and which ones compared? Which control variables should be considered? Should specific measures be combined or transformed or both?"

Simmons, Joseph P., Leif D. Nelson, and Uri Simonsohn. "False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant." Psychological science 22.11 (2011): 1359-1366.

# Concluding

- Hypothesis testing is used quite extensively
- And abused more often
- Cross validation?
- Real life problems (usually) have more data and are more noisy
  - But you can send e-mails, get clicks etc. trivially
- If there is one think to keep from this lecture is the use of bootstrapping to learn parameter confidence intervals
  - We will use bootstrap later on this module when we are going to model things