CE888: Data Science and Decision Making
Lecture 2: Summary and resampling statistics

Ana Matran-Fernandez
(Slides adapted from Spyros Samothrakis)
University of Essex

January 21, 2019

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 (referred to as the sample)
  - Resampling statistics create new samples from the original one and allow us to gain further insights
- Resampling statistics are far more intuitive to understand than using t-tests (I think...)

An example problem

- Find the salaries of the employees of a business
- When you only have information about some employees (through friends and acquaintances)

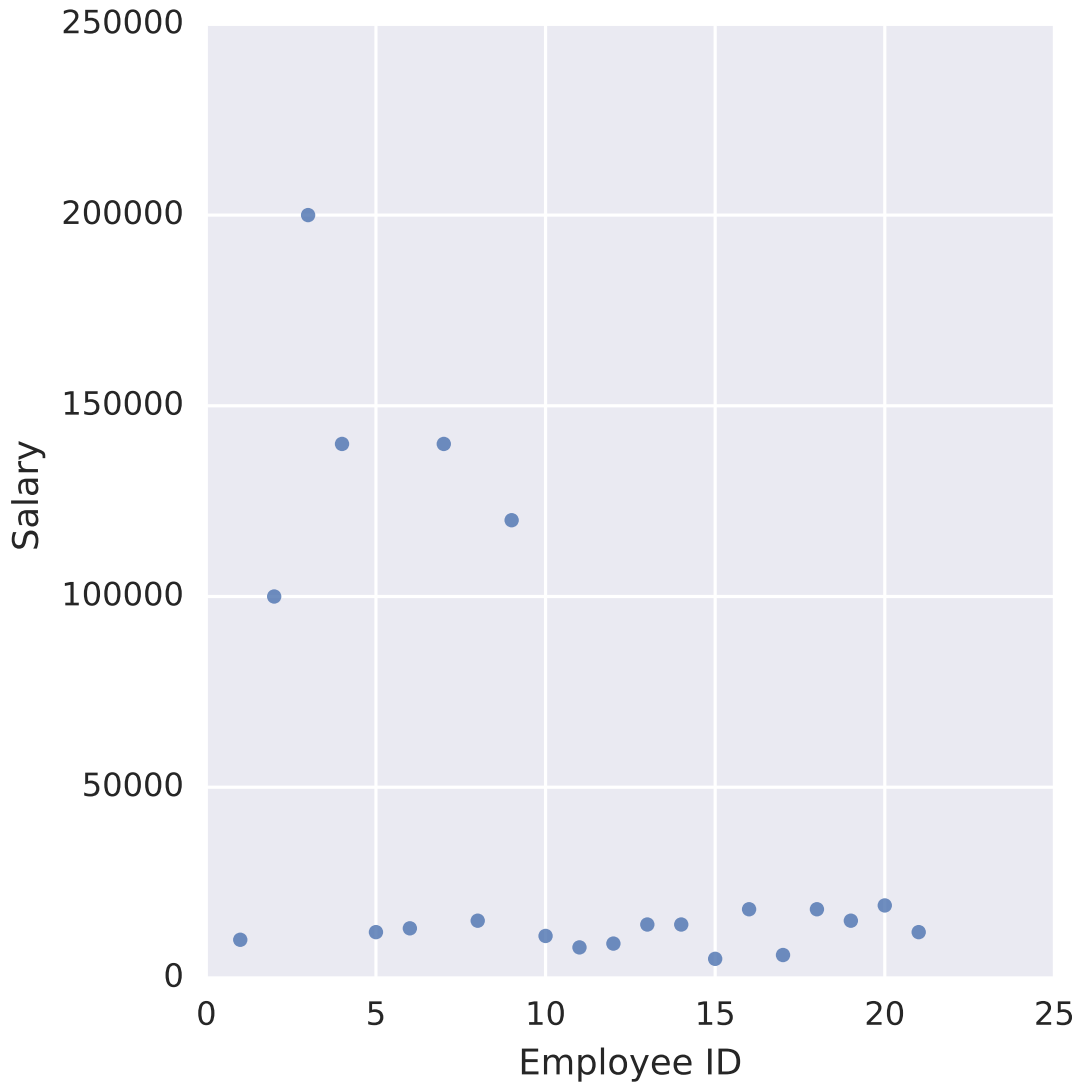
Employee ID	Salary
1	10000
2	100000
3	200000
4	140000
5	12000
6	13000
7	140000
8	15000
9	120000

(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

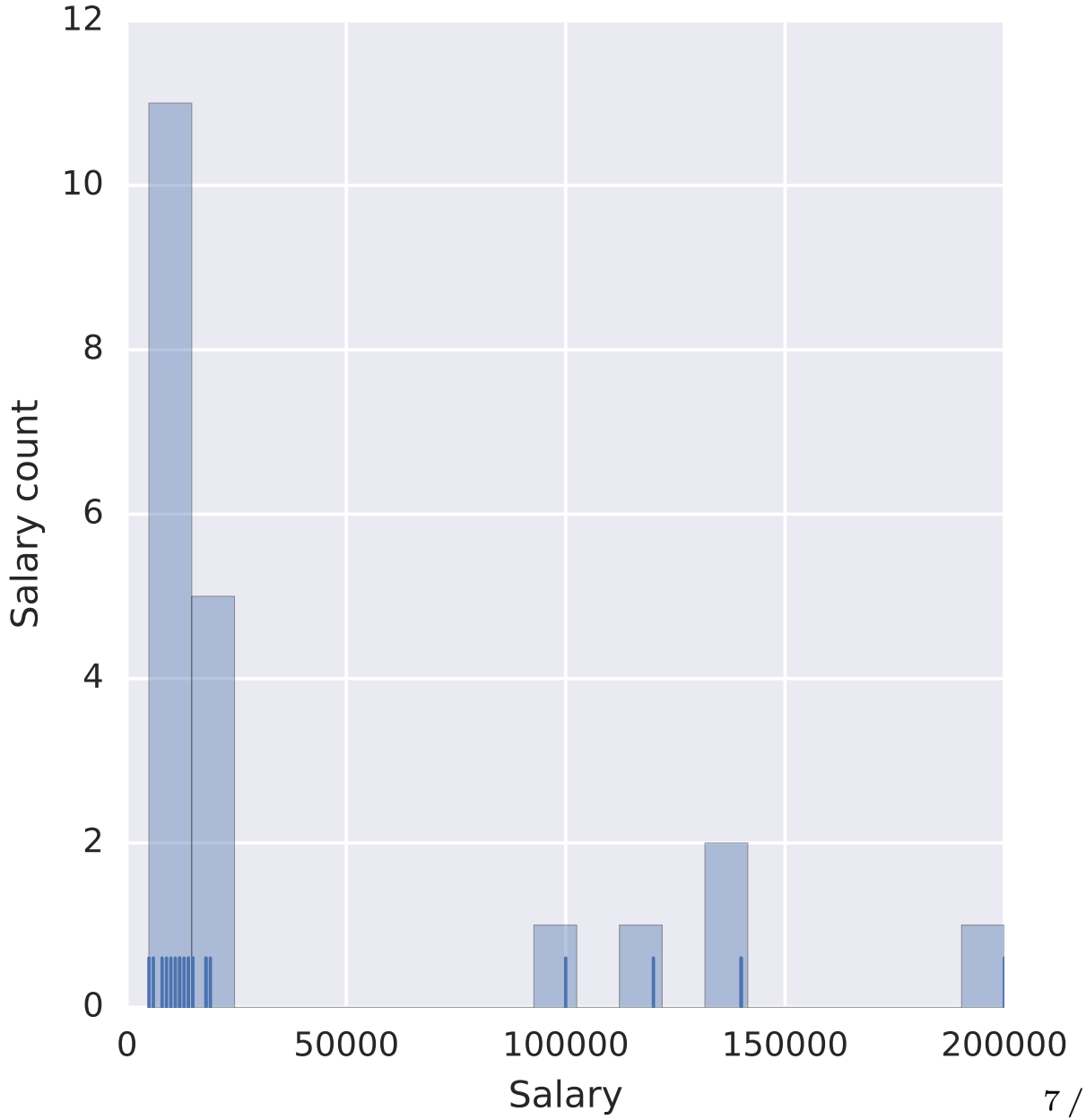
Visualising the data

```
import pandas as pd
import seaborn as sns

df = pd.read_csv('./salaries.csv')

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

sns_plot = sns.lmplot(df.columns[0]
df.columns[1],
data=df,
fit_reg=False).get_figure()
```



Histogram plot

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



Measures of central tendency

- (Sample) mean
  - $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
- (Sample) median
  - Rank x_i
  - $M = \begin{cases} x_{n/2+1} & \text{if } n \textit{ is odd} \\ (x_{/2} + x_{(n+1)/2})/2 & \text{if } n \textit{ is even} \end{cases}$
- In the salary sample:
  - $\bar{x} = 42809.523810$
  - $M = 14000.000000$

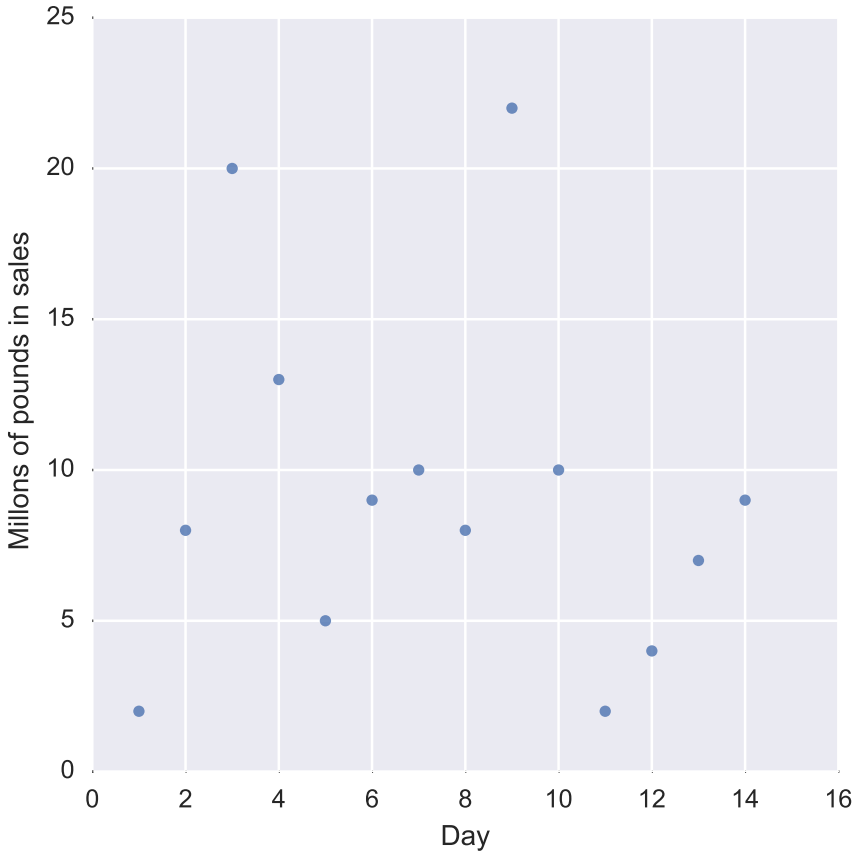
Measurements of dispersion

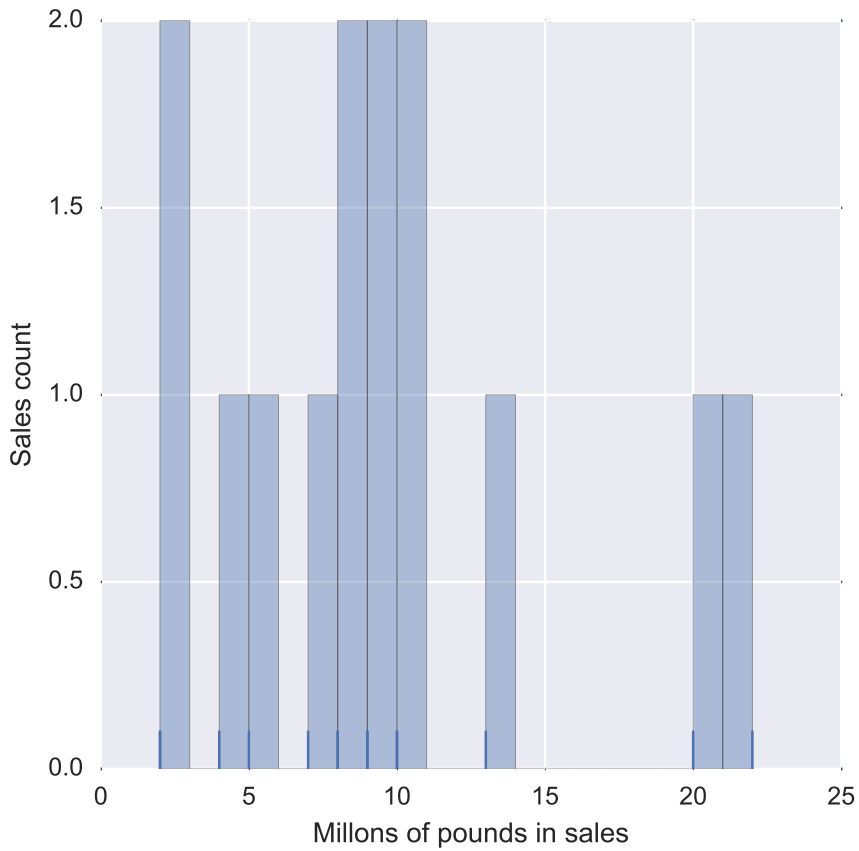
- (Sample) Standard deviation
  - $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$
  - Variance is s^2
- Median absolute deviation
  - $MAD = M(|x_i - M(x)|)$
- In our sample:
  - $s = 56841.147946$
  - $s^2 = 3230916099.773242$
  - $MAD = 4000.000000$

Sales dataset

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

Visualisation of Sales dataset





Summary statistics of the sales dataset

$\bar{x} = 9.214$

$M = 8.500000$

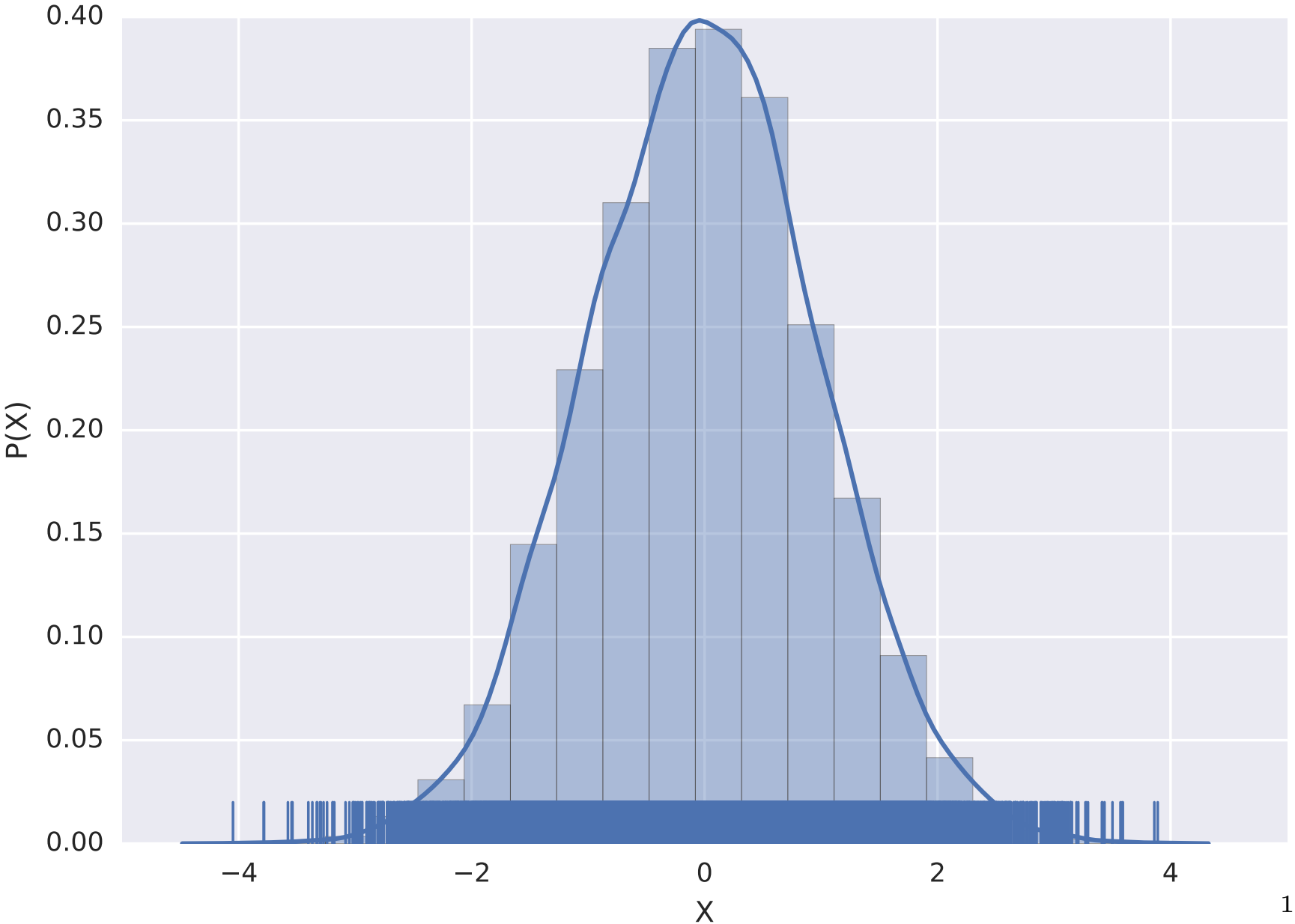
$s = 5.684296$

$s^2 = 32.311$

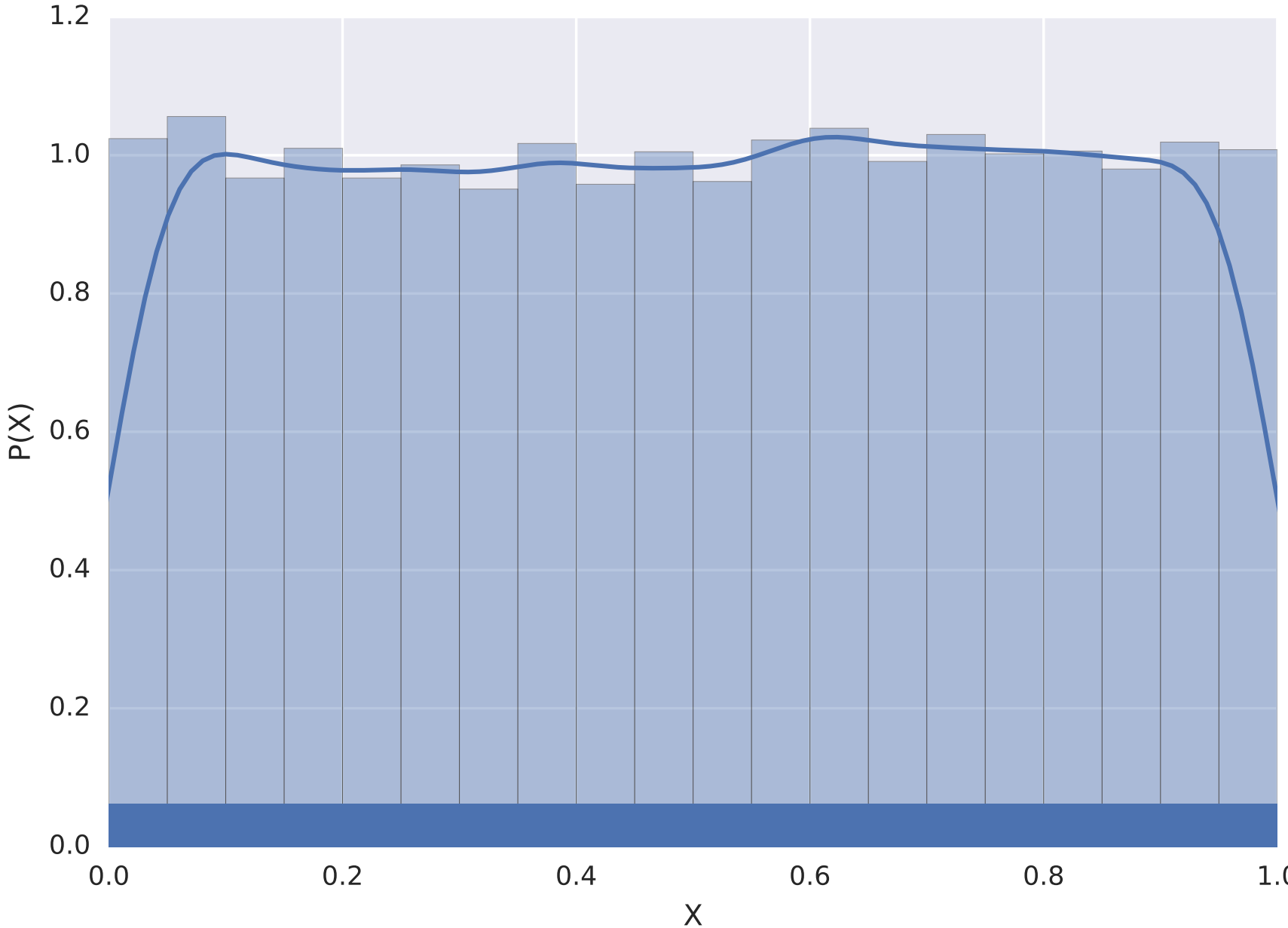
$M = 2.500$

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

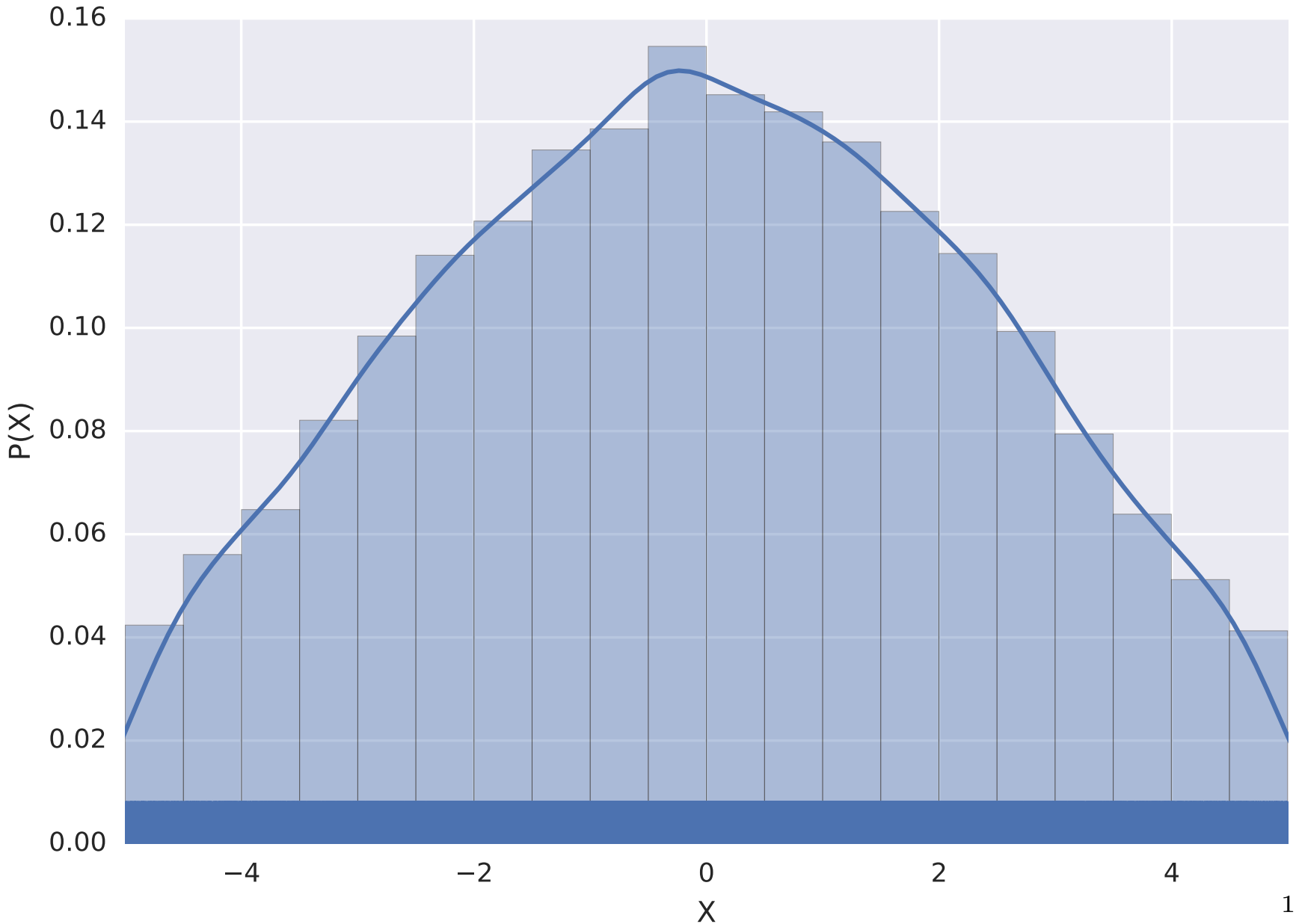
Normal distribution



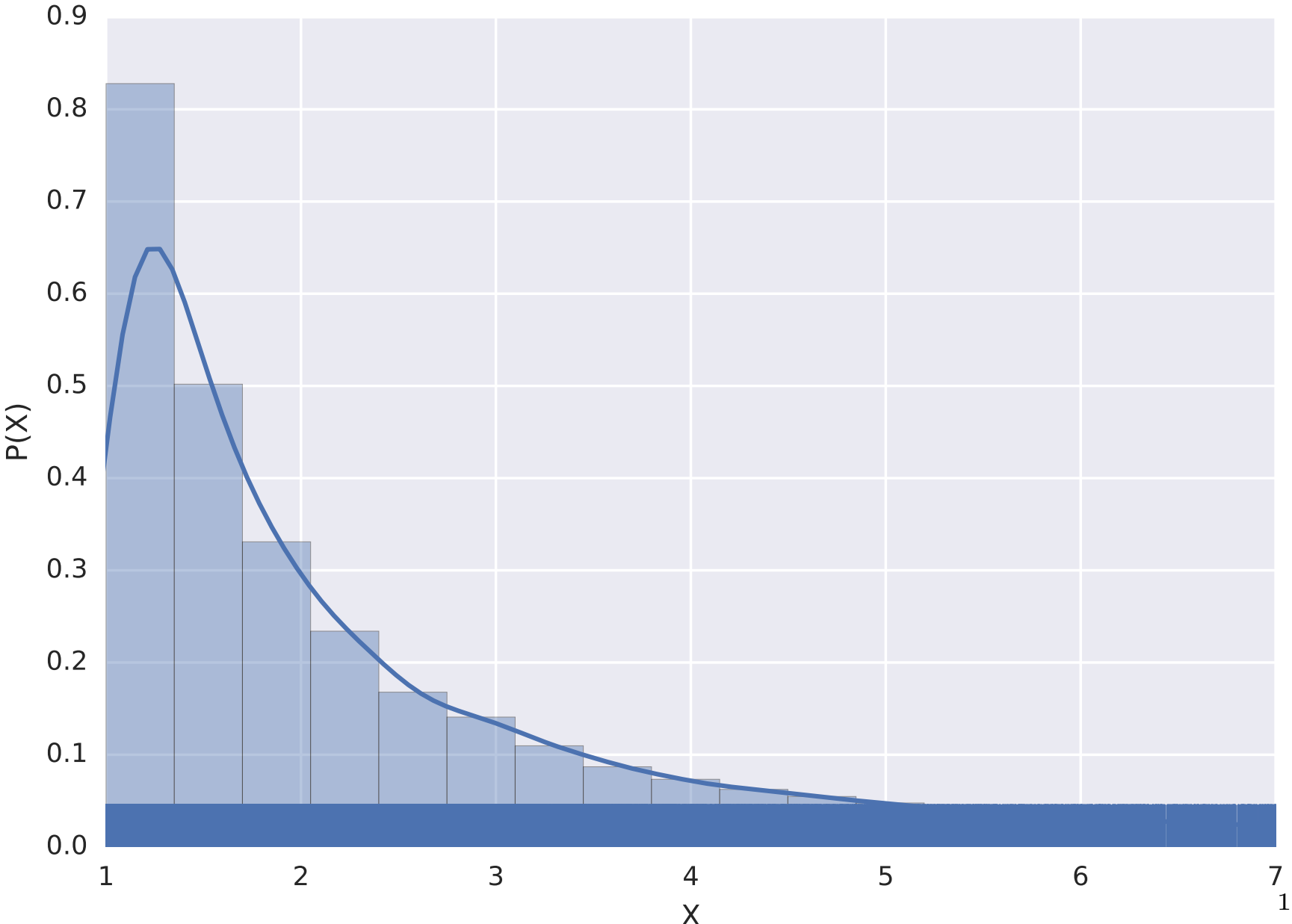
Uniform distribution



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 would like to build some notion of "confidence intervals" (CI)
  - Get a measure of "If I do this sampling process over and over again, what would I expect to be seeing?"
- 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"
- Hard to do without a machine

Bootstrapping (1)

- Ideally, we could possibly sample again and again from the real population
  - i.e. the journalist would go over to a different set of friends
  - Ask them to get more salaries
  - Repeat!
- Once we have a collection of different means we can say that the mean will fall within a certain range with a certain probability
  - But this is almost impossible (e.g., expensive, or not feasible)
- However, we can use our sample in a smart way
  - Resample from the sample!

Bootstrapping (2)

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

$x^1 = x_4^1, x_5^1, x_3^1, x_5^1...$

$x^2 = x_3^2, x_7^2, x_7^2, x_8^2...$

$x^{...} = ...$

$x^B = x_8^B, x_3^B, x_2^B, x_4^B...$

Bootstrapping (3)

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

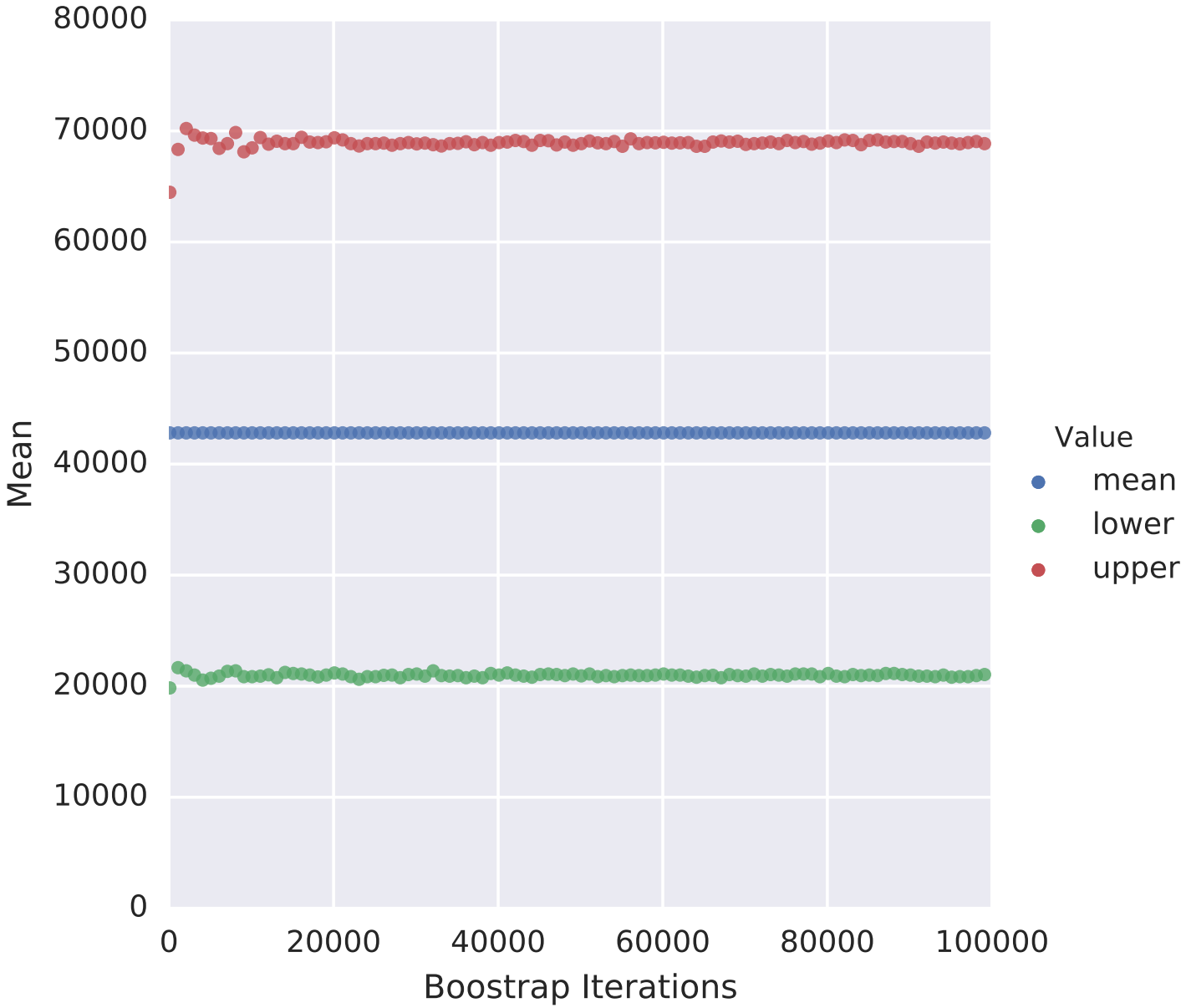
Bootstrapping (4)

- 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 sales data

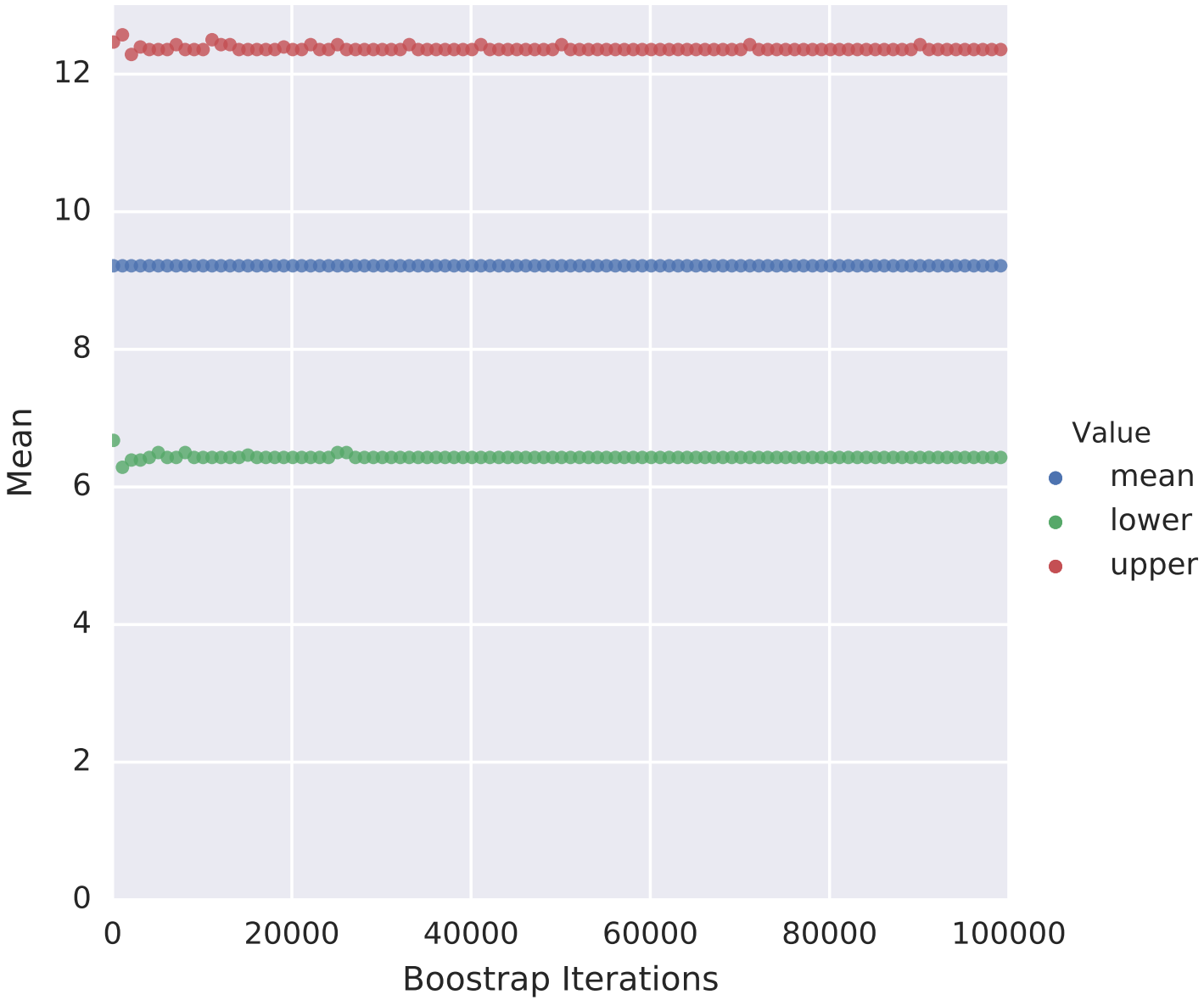
$$x = [6.86, 7.29, 7.86, 8.14$$
$$8.36, 8.79, 8.86, 9.14$$
$$9.29, 9.5, 9.5, 9.71$$
$$10.36, 11.14, 11.14, 13.21]$$

- 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

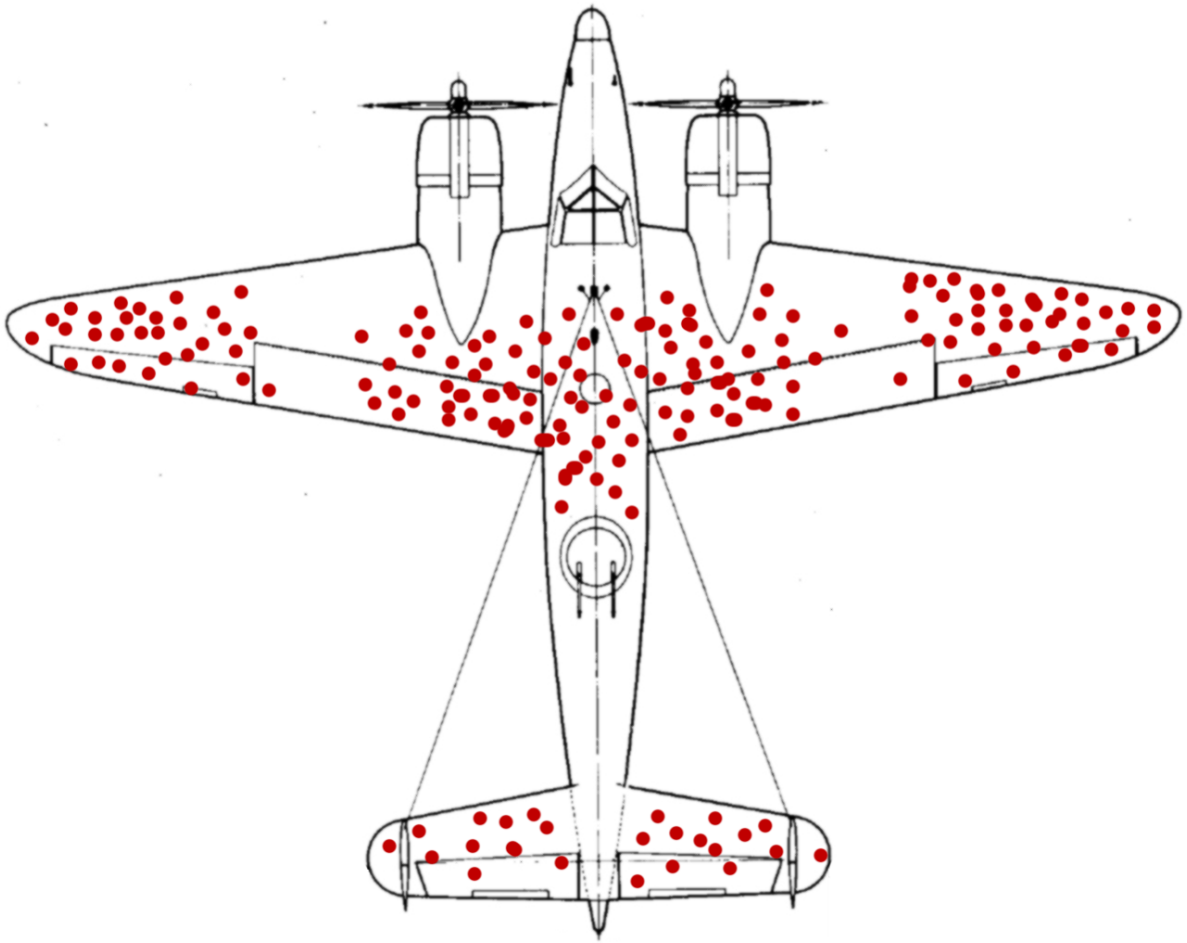
What can we say about the means now?

- 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

Data bias

- I have described a very biased process of collecting samples
  - 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 dataset?
  - Was there anything extraordinary on the day these measurements were taken?
  - Maybe it was Christmas
- Be very careful to randomise properly or at least make sure that you state your bias

Example: survivorship bias



[1] By McGeddon - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=53081927/

A/B Testing

- Suppose you had two versions of a website
  - and you would like to check if the newer version is better
- Two versions 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
- Not just limited to binary cases
- 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-imaginings”

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

$\bar{x}_{old} = 0.18$

$\bar{x}_{new} = 0.455$

$t_{obs} = \bar{x}_{new} - \bar{x}_{old} = 0.27$ (observed value of the test statistic)

Should they change to the new design?

Hypothesis forming

H_0: The two e-mails have 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 - alternative hypothesis

- Set significance level $\alpha = 0.05$, or equivalently, check if the 95% CI of t_{obs} under H_0 does not contain H_1
- p value = What is the probability of observing something as extreme as what we just observed by pure chance?

Permutation testing (1)

- Merge all the data into a new array (remember array concatenation using numpy from the first lab!)

```
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 randomly, 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 (first half and second half)

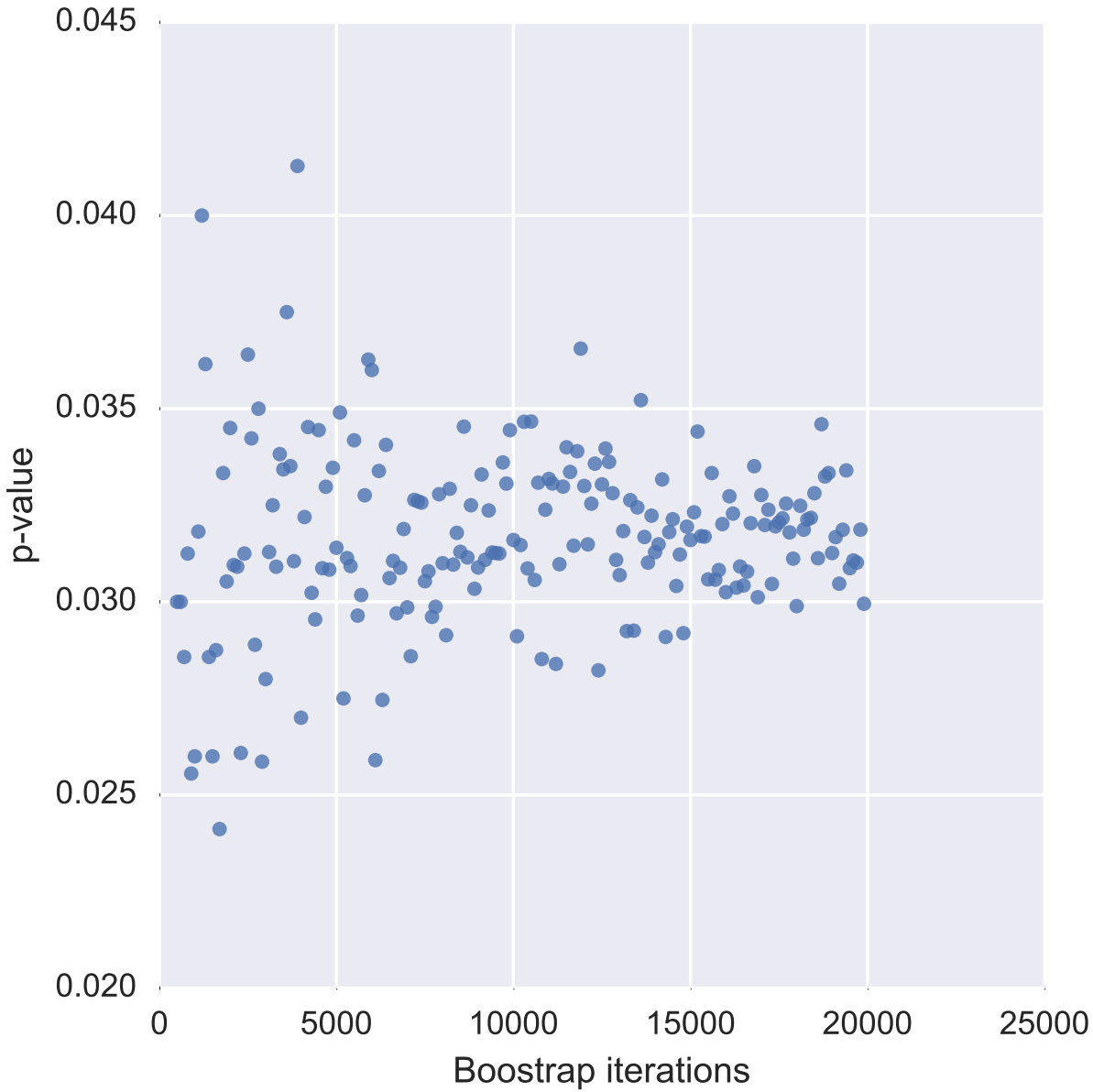
```
pold = np.array([0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1])
pnew = np.array([0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0])
```

- Record if the value of the test was more extreme or not
  - $t_{perm} = \bar{x}_{pnew} - \bar{x}_{pold}$
  - $t_{perm} > t_{obs}$
- 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 repeat this process 19,000 times you get $p = 0.032$
- Hence we can conclude 3% of the time you will get a higher difference in means than t_{obs}
- Since this number is smaller than our 5% significance level, you can reject the *null* hypothesis H_0
- So we conclude that the new format is better

Permutation test (4)



Another experiment

- Bob decides that adding a sound to the e-mail should increase user clicking even more
- Thinking that 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])
```

$\bar{x}_{old} = 0.546$

$\bar{x}_{new} = 0.73$

$t_{obs} = \bar{x}_{new} - \bar{x}_{old} = 0.19$

Results

- With $19,000$ permutations we get $p = 0.07$
- Thus, we have failed to reject the null hypothesis
- It does not mean that the sound does not 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

- α = significance level of our test, but also...
- α = Probability of Type I error, i.e., False positive rate
- Specificity = $1 - \alpha$ = the proportion of true negatives
- The lower the specificity, the more susceptible the test is to Type I errors
- Think of this as raising false alarms

Sensitivity

- False negative rate refers to another parameter called β, the probability of a Type II error.
- Sensitivity $= 1 - \beta =$ Power of a test, or the ratio of true positives
- The higher the sensitivity (i.e., the smaller β), the less we are bound to do Type II errors
- Think of this as failure to detect a phenomenon
- It is indirectly influenced by effect size and sample size
- Power = the ability of a test to detect a specific effect, if that specific effect actually exists.
- "Surely you only need one of them!" (No!)

α / β TRADE-OFF

- E.g., Increasing the significance level from 5% to 10%
- Where before we looked at 95% CI, now it's 90% CI
- i.e: Is my p value < 0.1?
- Increases the chance of rejecting H_0 when H_0 is false (i.e., reduce type II error; FN)
- But increases risk of statistical significance (i.e., reject H_0) when H_0 is not false (i.e., increase type I error; FP)

Power analysis (1)

- 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
  - You don't know the effect size you are testing for!
- In practice, sample as much as you can
- See previous studies in the literature
- If you have done a study before, use the boostrap!
- You might be tempted to increase α, but this will increase your chance for a Type I error

Power analysis (2)

- $power = P(reject\ H_0 | H_1\ is\ true)$
- Since it's a probability, it ranges between 0 and 1
- Higher power = lower probability of a type II error
- P = $1 - \beta$, where β is the probability of your test getting a type II error

Power analysis - example

- Power analysis can be used to calculate the minimum sample size required so that one can be reasonably likely to detect an effect of a given size.
- **How many samples do I need to in order to distinguish a 10-sided die from 20-sided die with $CI = 95\%$**
- **Power value of 0.9**
- Monte Carlo to the rescue
  - Gather data from your processes randomly
  - Calculate the p-values
- Code it!

A more "hack-ish 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 thing to keep from this lecture is the use of bootstrapping to learn parameter confidence intervals
  - We will use the bootstrap later on this module when we model things