The Inspection Paradox is Everywhere

Allen Downey
Olin College

PyData

Go to greenteapress.com/ip and follow instructions.

Experiment

Suppose you want to know average family size in the U.S.

And you have a convenience sample.

https://twitter.com/CambridgeSpark/status/1053322495267356673

Could ask, "How many children do you have?"

But the respondents are young.

And we want complete family sizes.

Idea!

Go up a generation.

"How many children does your mother have?"

greenteapress.com/ip

Survey

Go to
greenteapress.com/ip
and do the survey.

Survey of Family Size

An attempt to estimate the average family size in the U.S. by asking adults how big their families are.

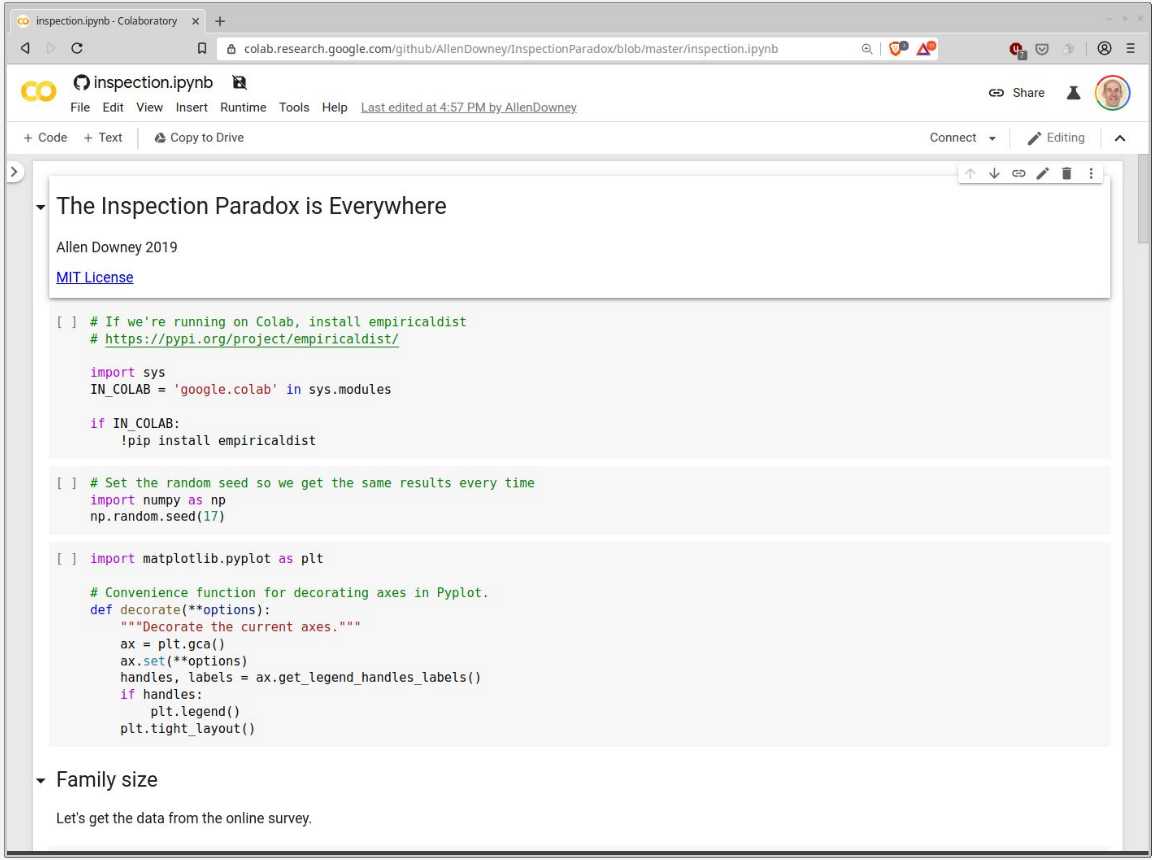
* Required

How many children has your biological mother born that were alive at birth? *

Your answer

Submit

When you are done, follow the link to the Jupyter notebook.



greenteapress.com/ip

Let's see some results.

Why might the PyData audience come from big families?

- Education
- Affluence
- Race and ethnicity
- etc.

Lots of possible sampling bias.

And one more thing...

Families with no children are not represented.

Families with no children are not represented.

Families with many children are over-represented.

Families with no children are not represented.

Families with many children are over-represented.

In general,

families with x children are over-represented by a factor of x.

Length-biased sampling

Sampling process where members of the population are sampled in proportion to size, length, duration, etc.

Inspection paradox

Subtly different sampling processes yield surprisingly different results.

Inspection paradox

- Common error, but not well known.
- Once you know about it, you see it everywhere.
- Often problematic, but sometimes useful for experimental design.

Average class size

Ask teachers how big their classes are. Average = 31

Ask students how big their classes are. Average = 56

Who’s lying?

Both right

They are averages across different populations.

Distribution of Undergraduate[1] Classes by Course Level and Class Size

(for Fall 2012)

Download a PDF of this page (Adobe Acrobat Reader Required).

	Class Size								
Course Level	**1**	**2-9**	**10-19**	**20-29**	**30-39**	**40-49**	**50-99**	**100+**	**Total**
000-199	38	164	659	917	241	70	99	123	2,311
200-299	82	108	370	486	307	84	109	134	1,680
Lower Level	120	272	1,029	1,403	548	154	208	257	3,991
Percent of Lower Level Total	3.0%	6.8%	25.8%	35.2%	13.7%	3.9%	5.2%	6.4%	100.0%
300-399	4	148	387	314	115	96	186	53	1,303
400-499	14	132	256	190	83	67	64	17	823
Upper Level	18	280	643	504	198	163	250	70	2,126
Percent of Upper Level Total	0.8%	13.2%	30.2%	23.7%	9.3%	7.7%	11.8%	3.3%	100.0%
500-599	0	79	102	67	43	29	23	2	345
600-699	0	4	14	5	7	8	6	4	48
800-899	0	0	0	0	0	0	0	0	0
Dual Level	0	83	116	72	50	37	29	6	393
Percent of Dual Level Total	0.0%	21.1%	29.5%	18.3%	12.7%	9.4%	7.4%	1.5%	100.0%
Total All Classes	138	635	1,788	1,979	796	354	487	333	6,510
Percent of All Classes	2.1%	9.8%	27.5%	30.4%	12.2%	5.4%	7.5%	5.1%	100.0%

[1]"Undergraduate" Classes refers to organized classes with one or more undergraduate students enrolled.

Average across students

138 classes with 1 student = 138 students

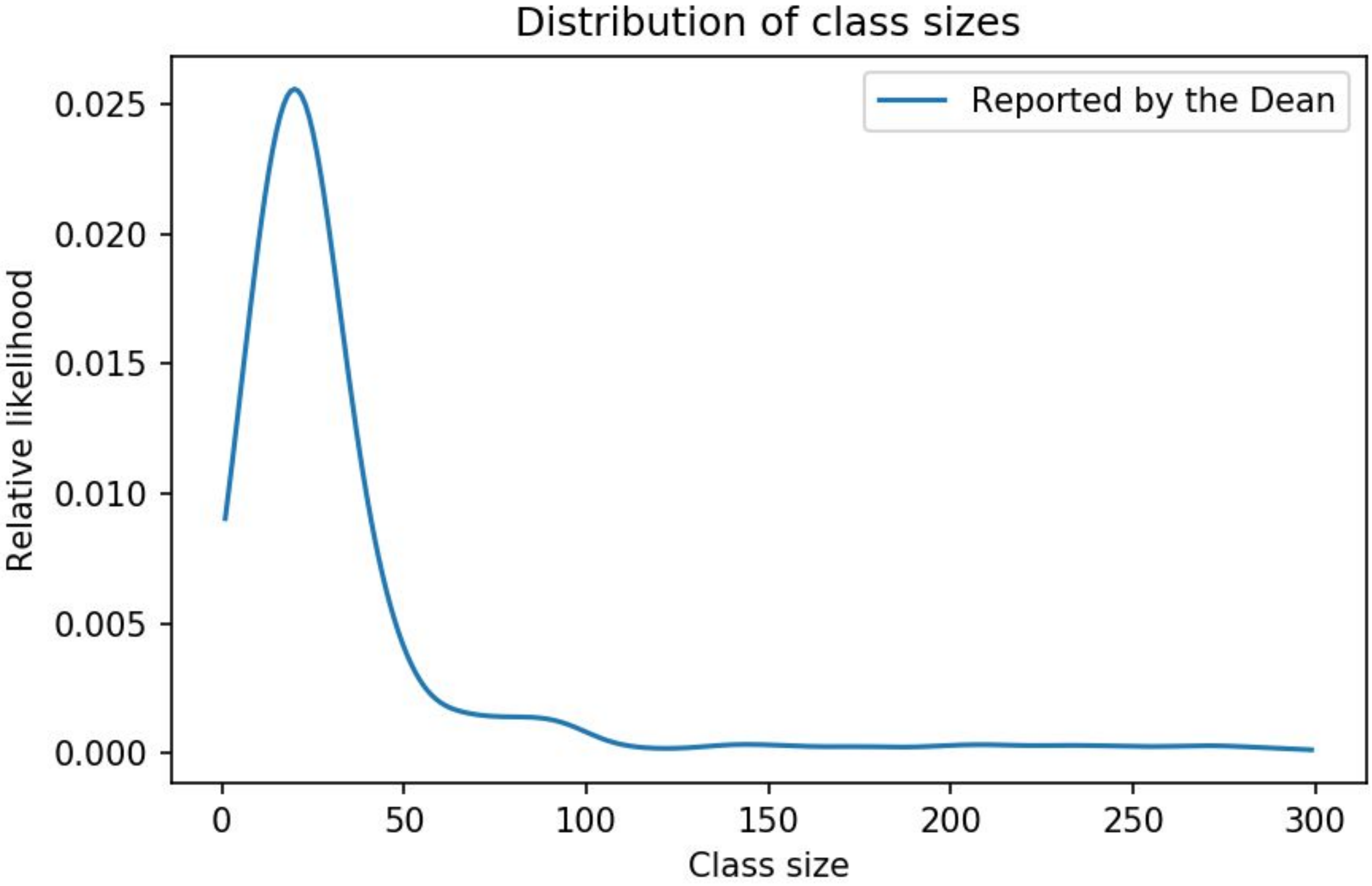
333 classes with 100+ students = 33,300+

Large classes get oversampled.

Class size x gets oversampled by x.

I used the data in their table to generate an unbiased sample of class size.

And kernel density estimation (KDE) to plot the distribution.



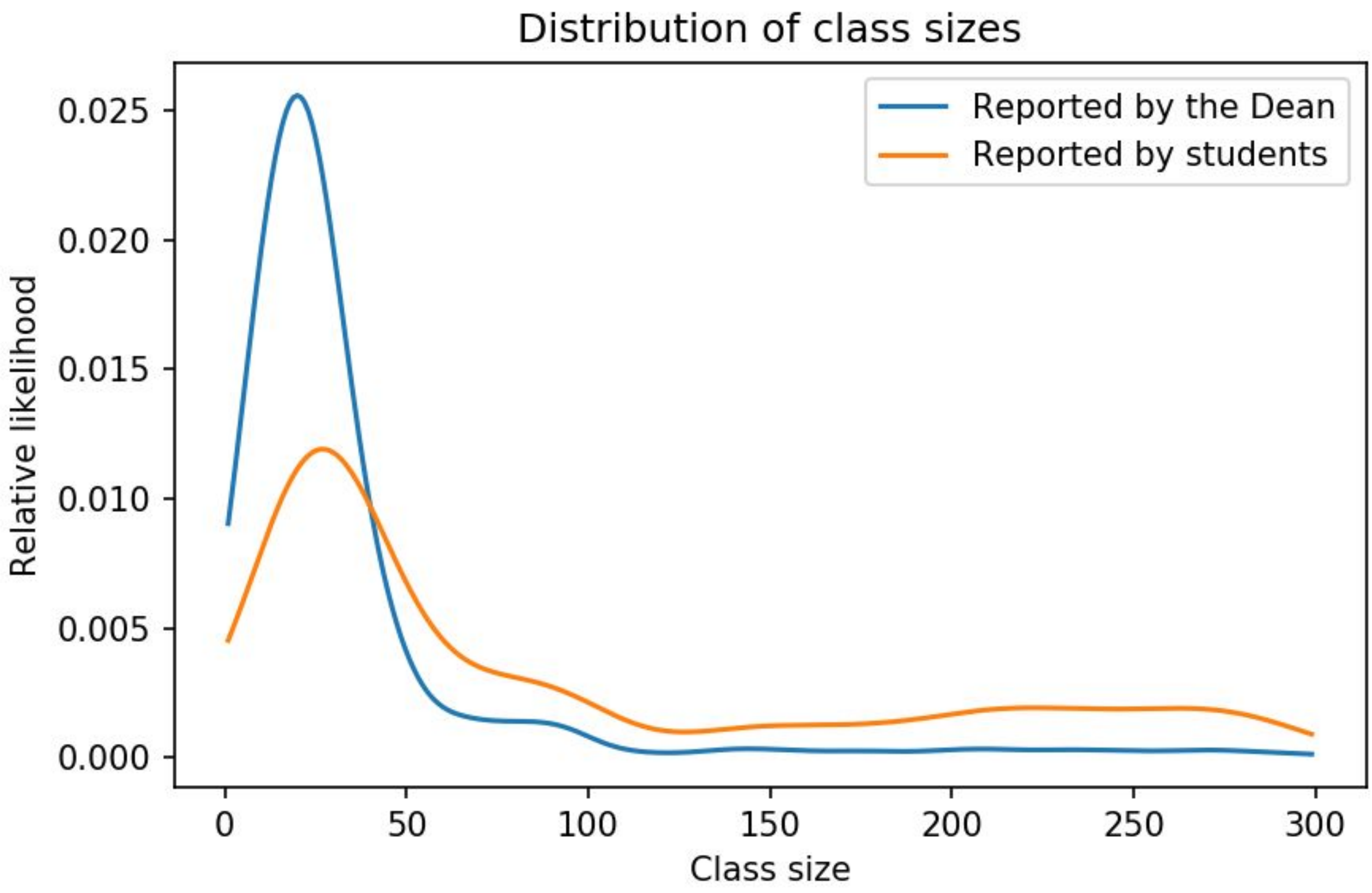
Distribution of class sizes
Reported by the Dean
Relative likelihood
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250
300
Class size

```python
def resample_weighted(sample, weights):
    """Generate a biased sample.

    sample: NumPy array
    weights: NumPy array

    returns: NumPy array
    """
    n = len(sample)
    p = weights / np.sum(weights)
    return np.random.choice(sample, n, p=p)
```

```python
biased = resample_weighted(unbiased, unbiased)
```



Distribution of class sizes
Reported by the Dean
Reported by students
Relative likelihood
0.000
0.005
0.010
0.015
0.020
0.025
0
50
100
150
200
250
300
Class size

If you are not careful,
this kind of biased sampling is a problem.

If you are clever, you can use it.

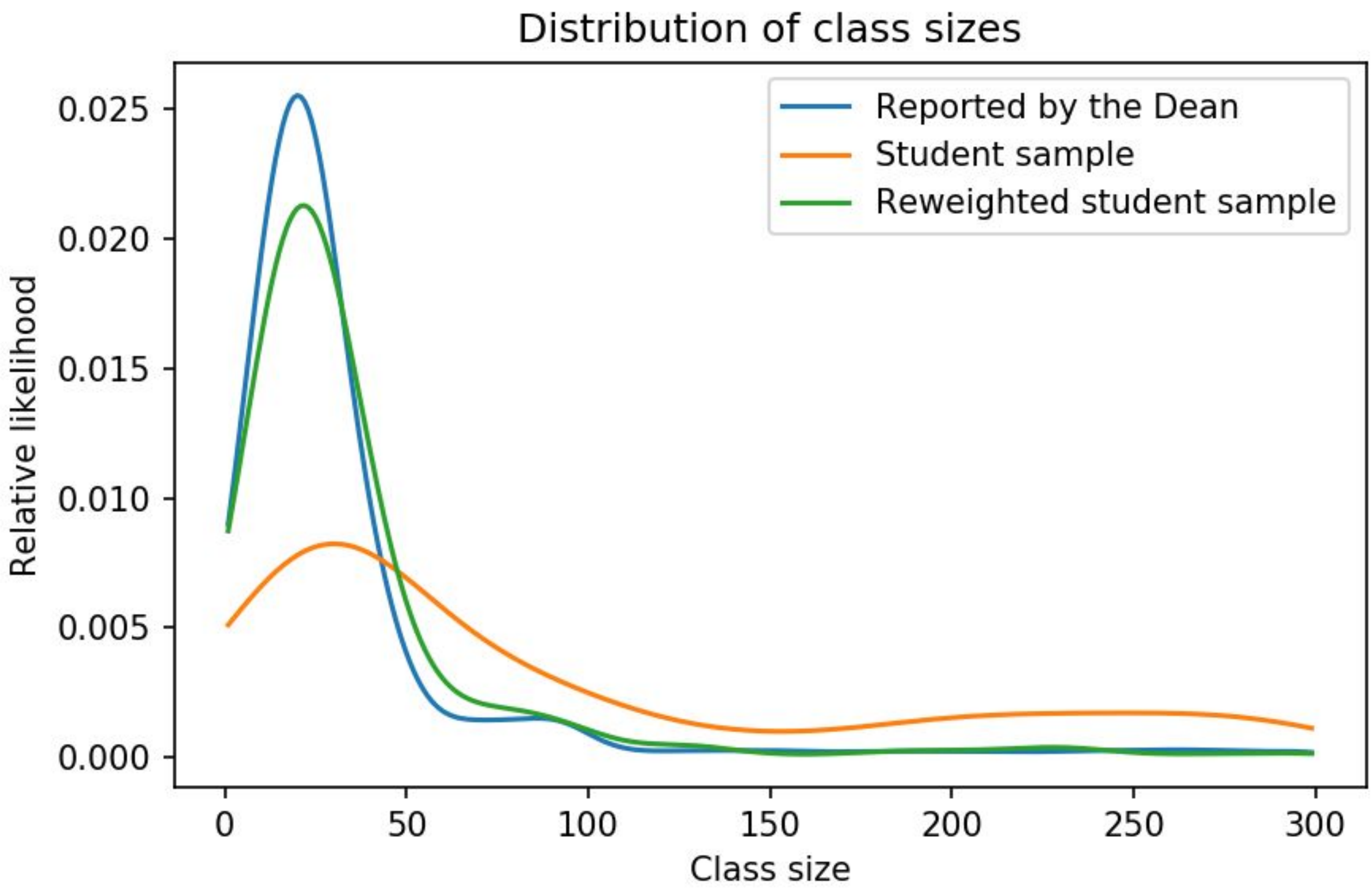
Suppose your school doesn't publish class sizes...

Experimental design

1. Sample students.
2. Resample with inverse weights.

```
sample = np.random.choice(biased, 500)
```

```
reweighted = resample_weighted(sample, 1/sample)
```


Distribution of class sizes
Reported by the Dean
Student sample
Reweighted student sample
Relative likelihood
0.000
0.005
0.010
0.015
0.020
0.025
0
50
100
150
200
250
300
Class size

It’s everywhere

Airlines:
“We are losing money because too many planes are nearly empty.”

Passengers:
“Flying is miserable because the planes are always full.”

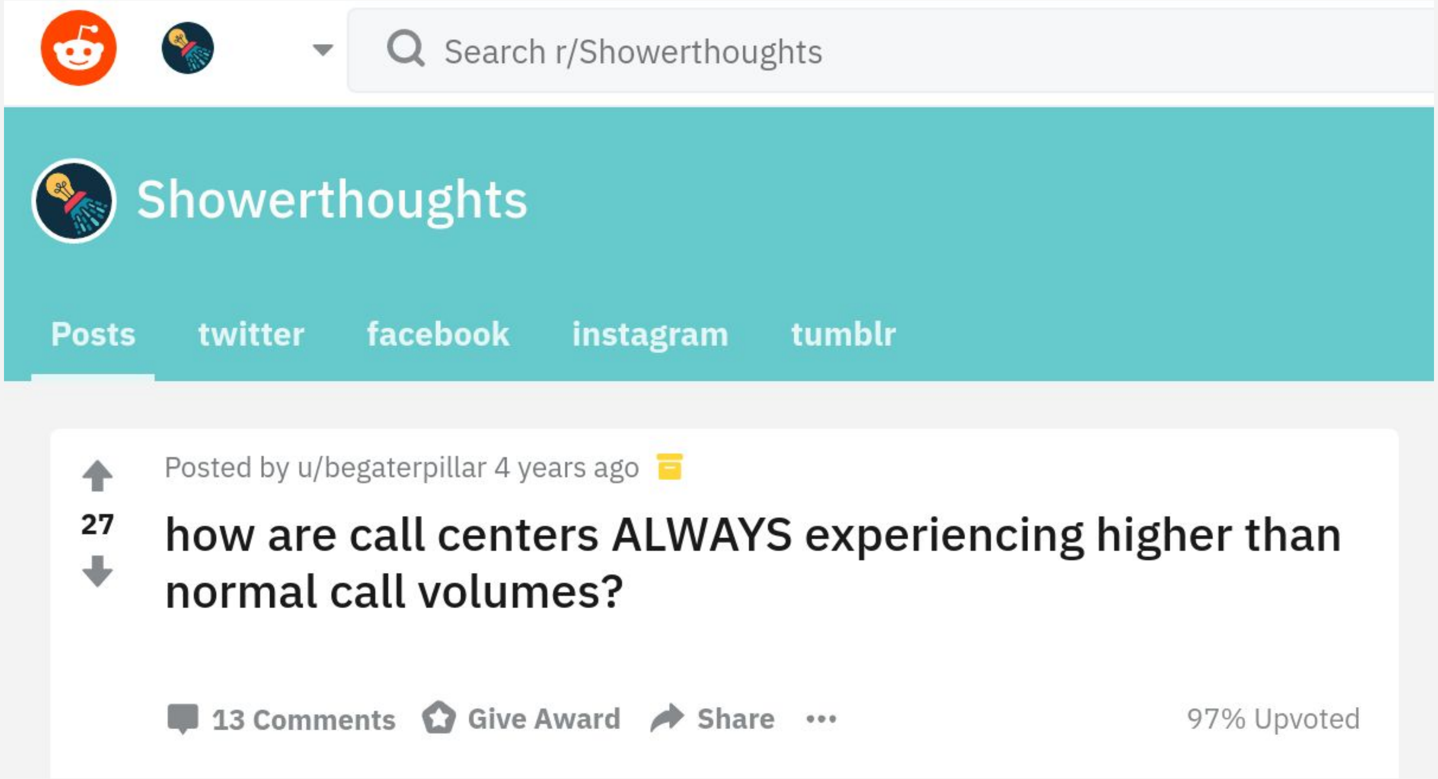
Both could be true

Few people enjoy a near-empty flight.

Lots of people suffer on a full one.

A plane with x passengers is oversampled by x.

It's everywhere



BUSINESS

Why You Can’t Get a Taxi

And how an upstart company may change that

MEGAN MCARDLE MAY 2012 ISSUE

WHERE I LIVE in Washington, D.C., about a mile and a half north of the Capitol, you can sometimes get a taxi in two minutes flat. And sometimes, after spending 20 minutes wistfully waving two fingers in the air while the traffic hurtles past, you have to give up and trudge to the train.

Waiting for the bus in the rain

Suppose buses run every 20 minutes on average.

You expect to wait 10 minutes, on average.

Right?

Nope.

If there's any variation,
there are long intervals and short ones.

You are more likely to arrive during a long one.

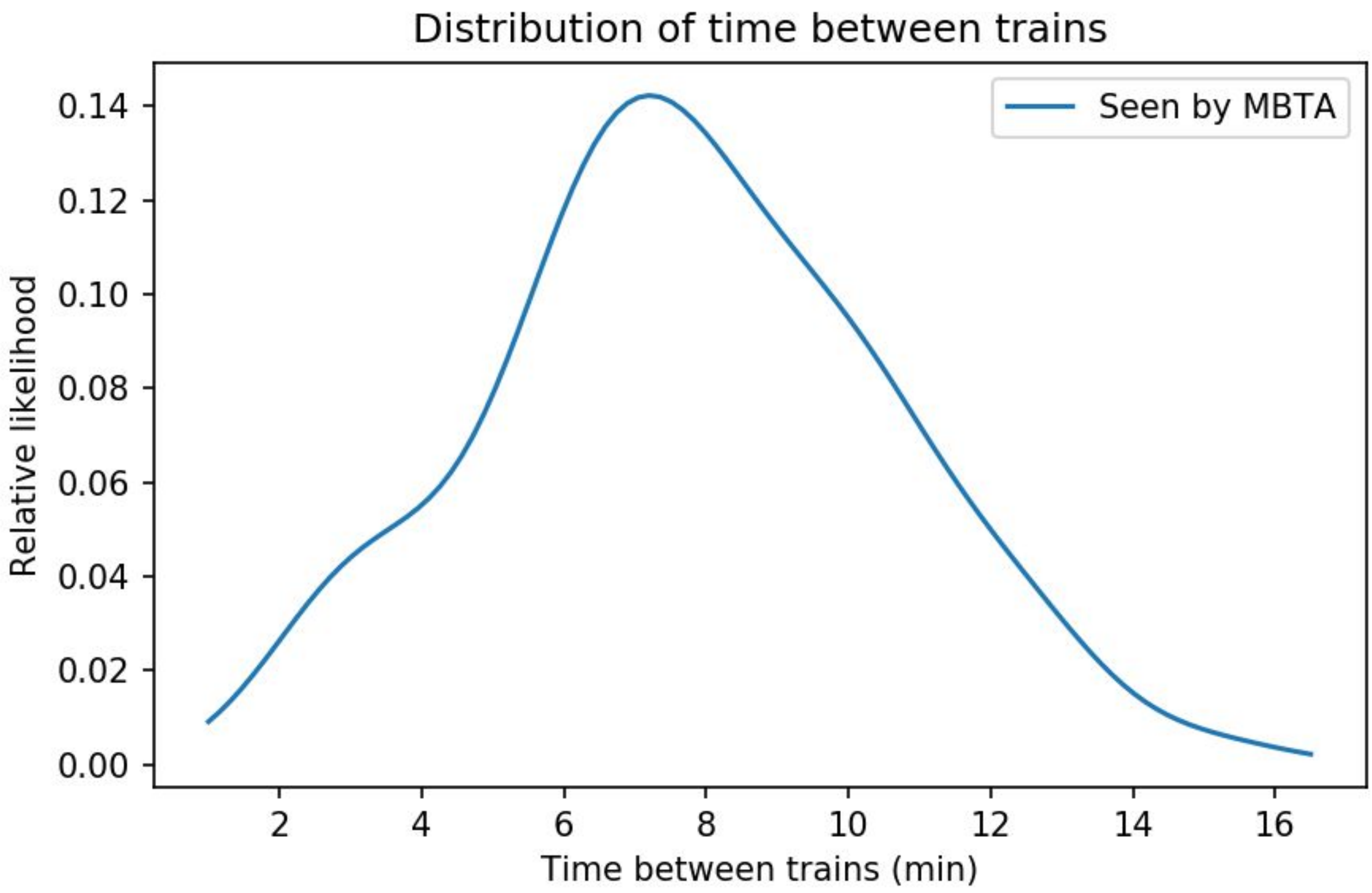
Nope.

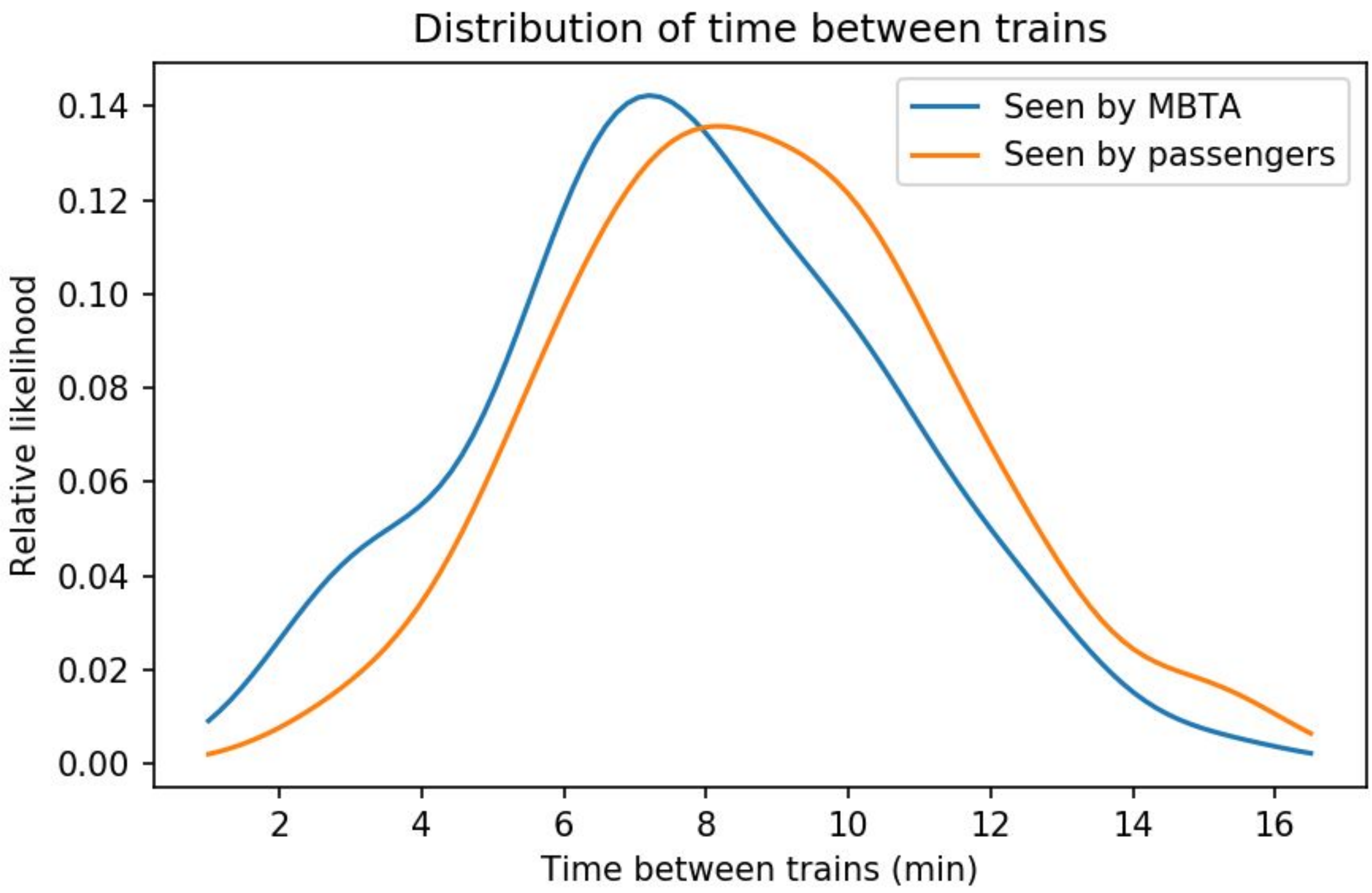
If there's any variation,
there are long intervals and short ones.

You are more likely to arrive during a long one.

Intervals of duration `t` are oversampled by a factor of `t`.

"Kendall Sq Inbound Platform" by Eric Kilby
https://commons.wikimedia.org/wiki/File:Kendall_Sq_Inbound_Platform.jpg


Distribution of time between trains
Seen by MBTA
Relative likelihood
Time between trains (min)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
2
4
6
8
10
12
14
16


Distribution of time between trains
Seen by MBTA
Seen by passengers
Relative likelihood
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
2
4
6
8
10
12
14
16
Time between trains (min)

Average reported by MBTA: 7.8 minutes.

Average observed by passengers: 8.8 minutes.

In this example the difference is moderate because the variance is moderate.

It can be much bigger.

Let me ask you a question...

Are you popular?

Hint: no.

Why You're Probably Less Popular Than Your Friends

Where averages and individual perspectives diverge

By John Allen Paulos | Jan 18, 2011

Are your friends more popular than you are? There doesn't seem to be any obvious reason to suppose this is true, but it probably is. We are all

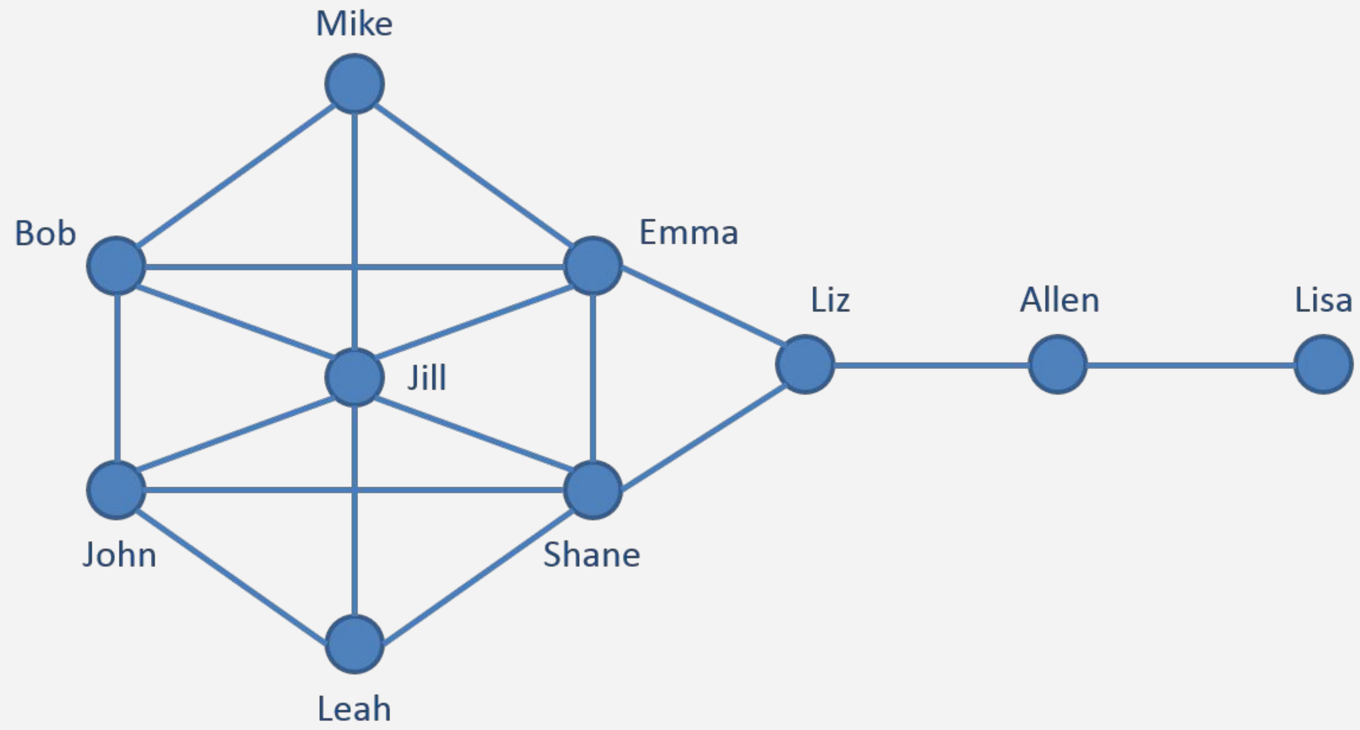
Friendship paradox

How many friends do you have on Facebook?

Suppose I pick one of your friends at random.

Chances are `~80%` that they have more friends than you.

Think of a graph



https://nauczanki.files.wordpress.com/2014/11/network.png

One more time

1. Choose a random node.
2. Choose a random edge and follow it.
3. Query that node.

A node with degree x is oversampled by x.

SNAP for C++
SNAP for Python
SNAP Datasets
What's new
People
Papers
Citing SNAP
Links
About
Contact us

Social circles: Facebook

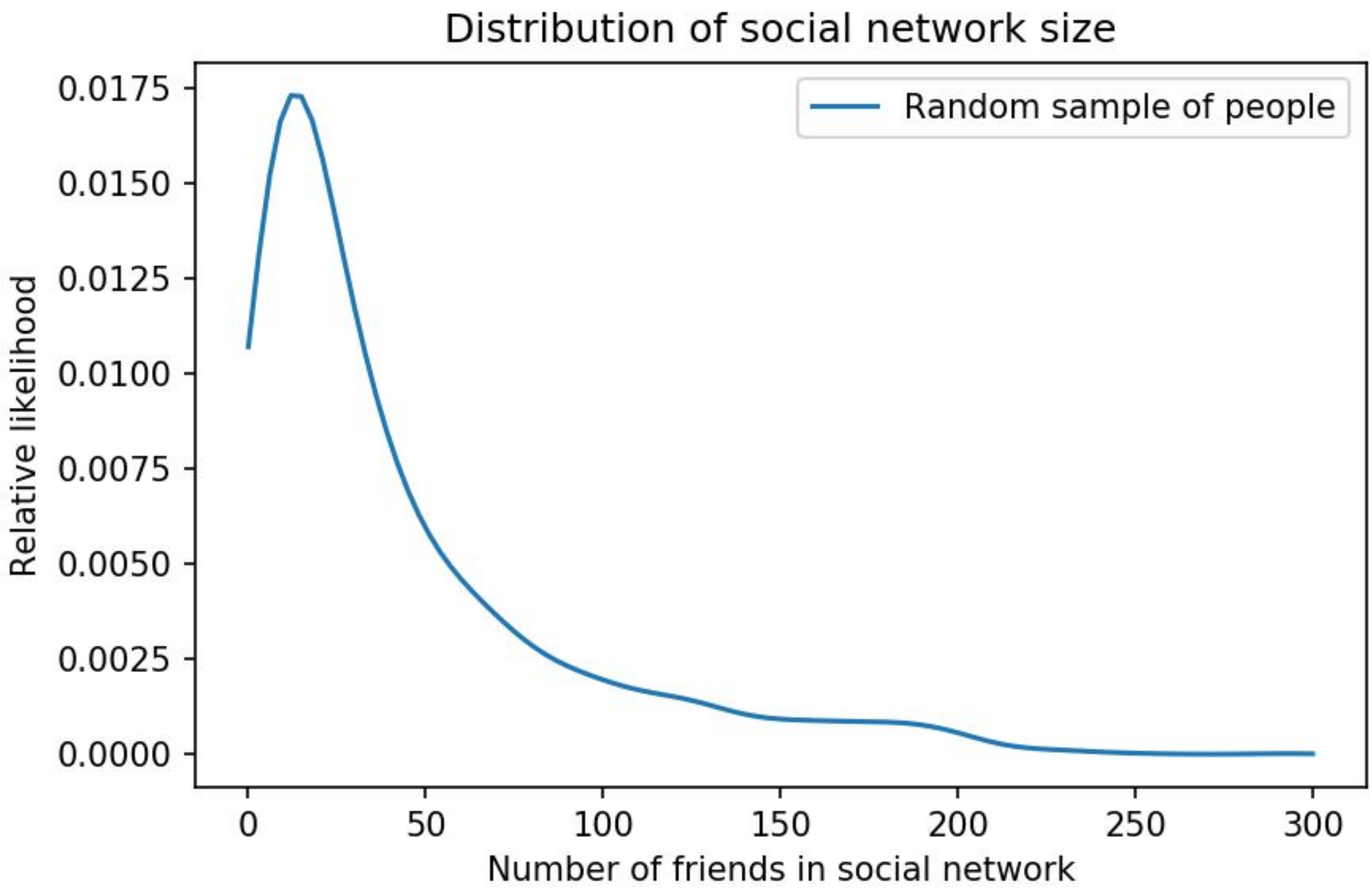
Dataset information

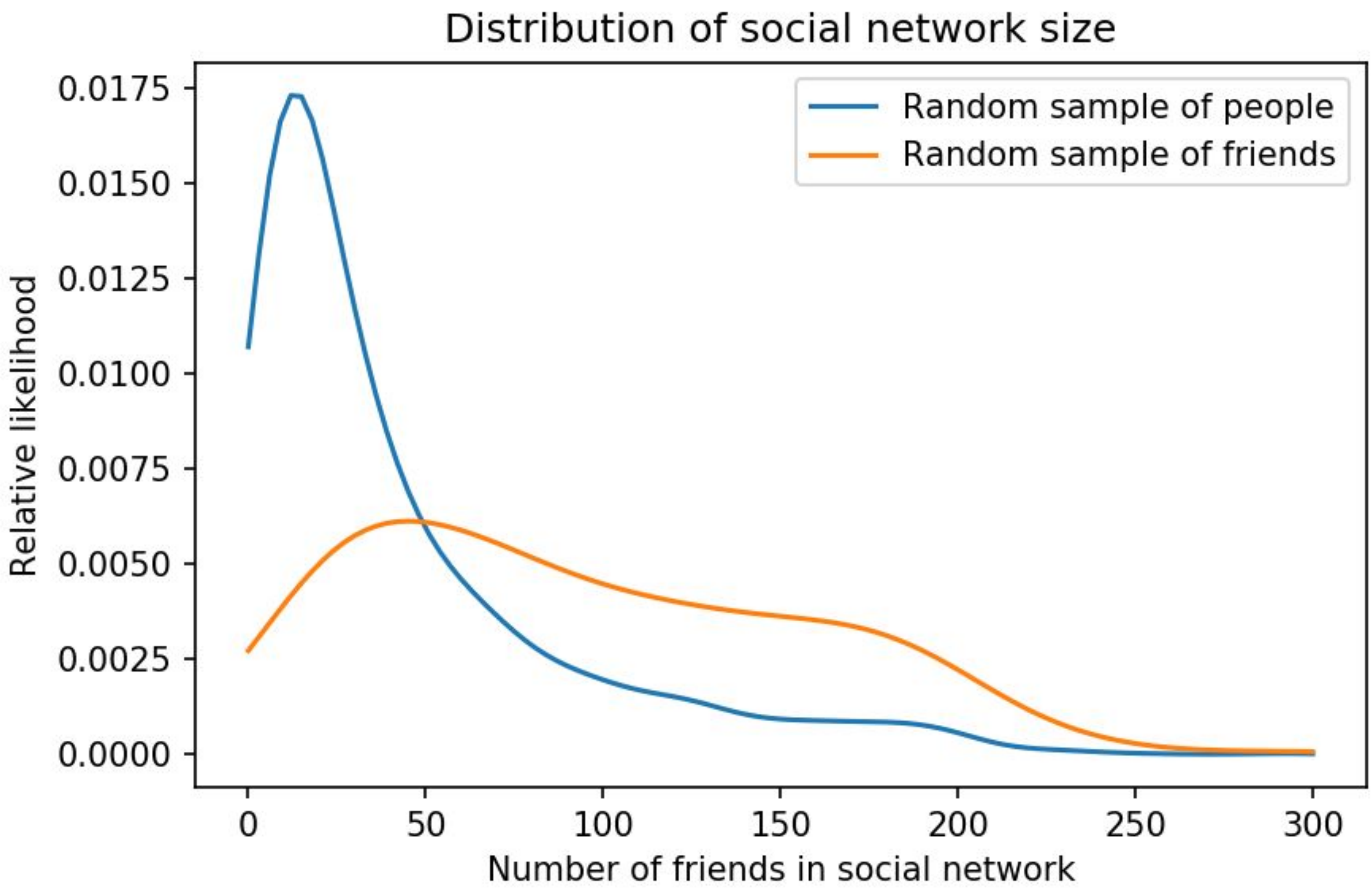
This dataset consists of 'circles' (or 'friends lists') from Facebook. Facebook data was collected from survey participants using this Facebook app. The dataset includes node features (profiles), circles, and ego networks.

Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. Also, while feature vectors from this dataset have been provided, the interpretation of those features has been obscured. For instance, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1". Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their individual political affiliations represent.

Data is also available from Google+ and Twitter.

Dataset statistics	
Nodes	4039
Edges	88234
Nodes in largest WCC	4039 (1.000)
Edges in largest WCC	88234 (1.000)
Nodes in largest SCC	4039 (1.000)
Edges in largest SCC	88234 (1.000)
Average clustering coefficient	0.6055
Number of triangles	1612010
Fraction of closed triangles	0.2647
Diameter (longest shortest path)	8
90-percentile effective diameter	4.7


Distribution of social network size
Random sample of people
Relative likelihood
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
50
100
150
200
250
300
Number of friends in social network


Distribution of social network size
Random sample of people
Random sample of friends
Relative likelihood
Number of friends in social network
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0
50
100
150
200
250
300

What a difference an edge makes

Average number of edges “you” have:

42

Average number of edges your friends have:

~100

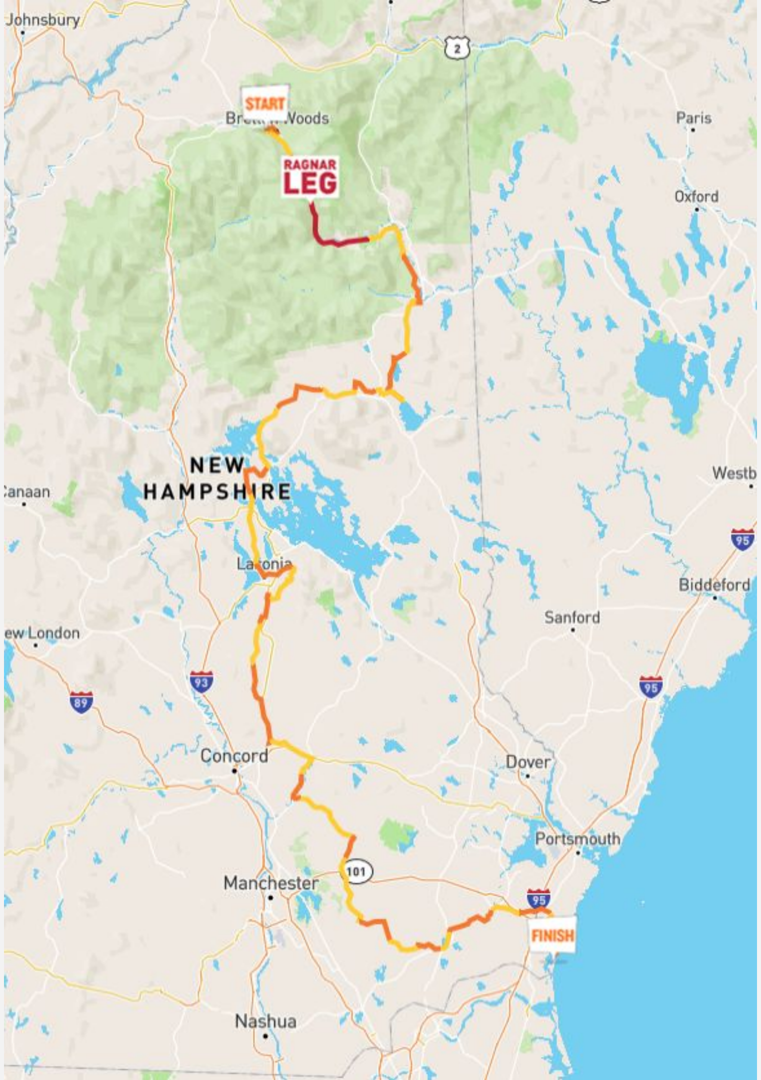
Reach the Beach

200+ mile relay.

When I overtook someone,
I overtook them fast.

When people overtook me,
they blew by me.

Bimodal distribution of runners?



Nope, it's the inspection paradox

Because of the format,
fast and slow runners are spread out.

Almost no relationship between speed and position.

As a runner, who do you see?

Fast runners pass many slow runners,
fewer fast ones.

Slow runners are overtaken by many fast runners,
fewer slow ones.

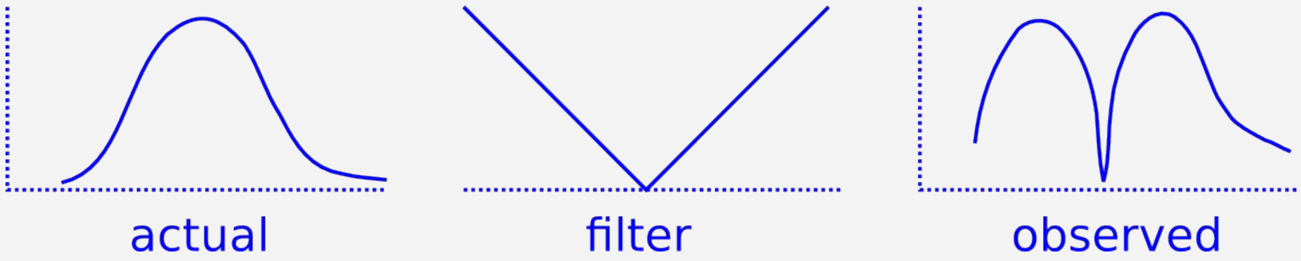
Fast runners pass many slow runners,
fewer fast ones.

Slow runners are overtaken by many fast runners,
fewer slow ones.

In general, the chance of seeing another runner is proportional to the difference in your speeds.

Prob of observing `x` is proportional to `abs(x-v)`

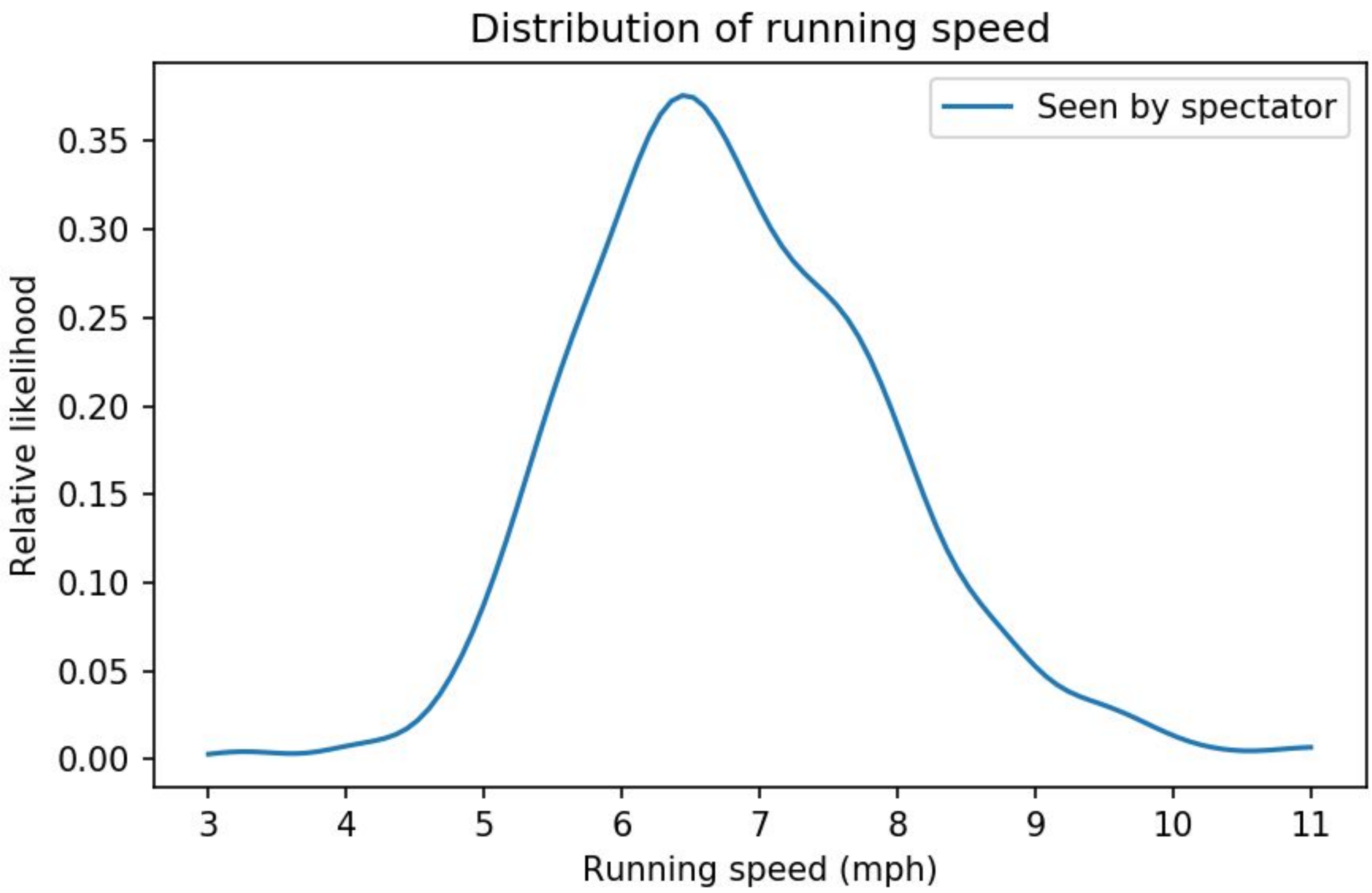
- Actual distribution is single-mode.
- Filter is v-shaped.
- Observed distribution is bimodal.



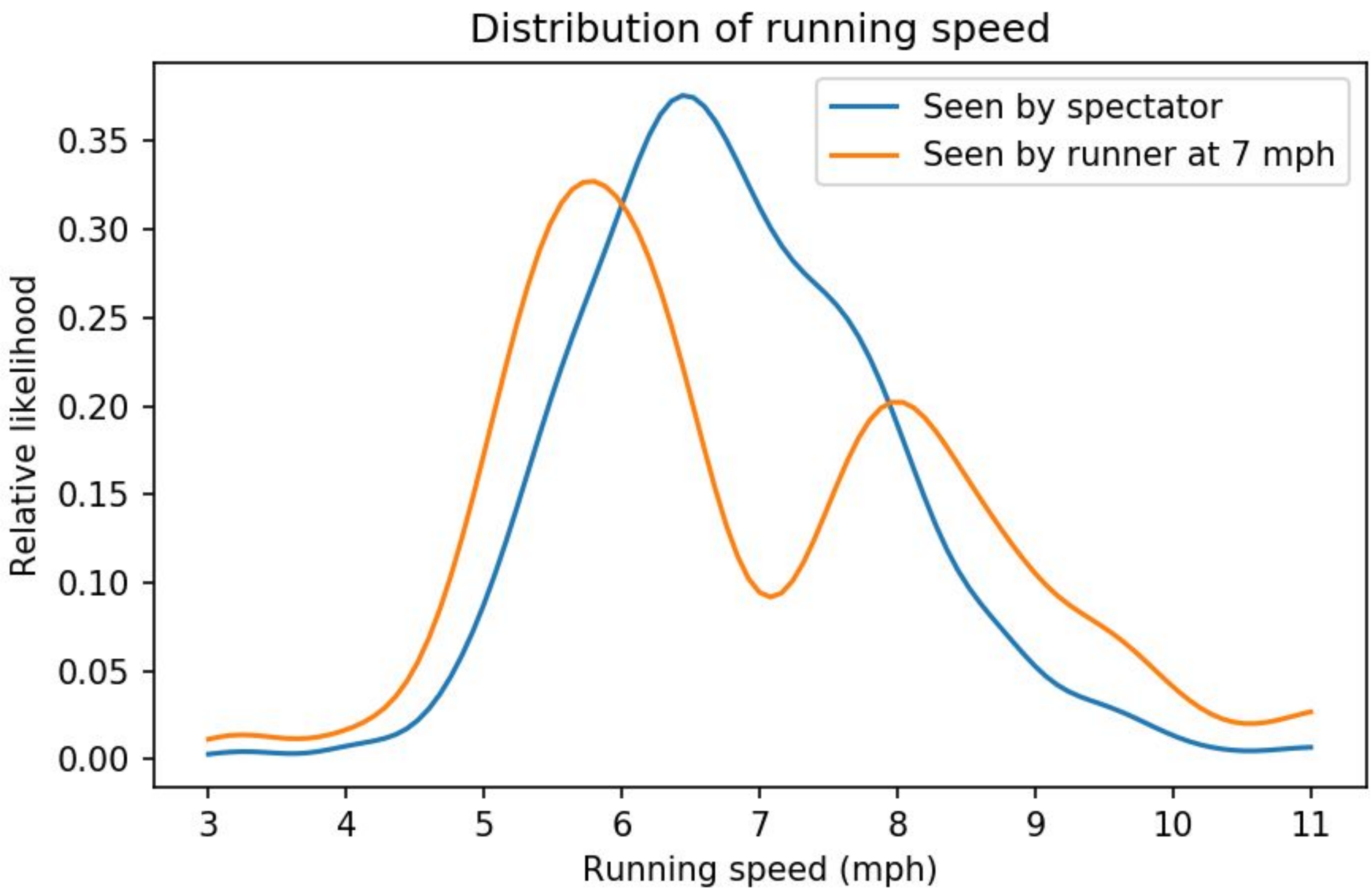
Data from a 10K road race.

JAMES JOYCE RAMBLE

Claim	Steven Kostick M 48 | Bib 1382 | MA	95	84	14	[illegible] min/mi	42:37
Claim	Steve Murray M 46 | Bib 1668 | VA	96	85	15	6:52 min/mi	42:39
AD	Allen Downey M 42 | Bib 337 | Needham, MA	97	86	12	6:53 min/mi	42:44
Claim	Kate Blake F 32 | Bib 107 | Dedham, MA	98	12	2	6:54 min/mi	42:48



Distribution of running speed
Seen by spectator
Relative likelihood
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
3
4
5
6
7
8
9
10
11
Running speed (mph)


Distribution of running speed
Seen by spectator
Seen by runner at 7 mph
Relative likelihood
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
3
4
5
6
7
8
9
10
11
Running speed (mph)

On the highway

Everybody is going too fast.

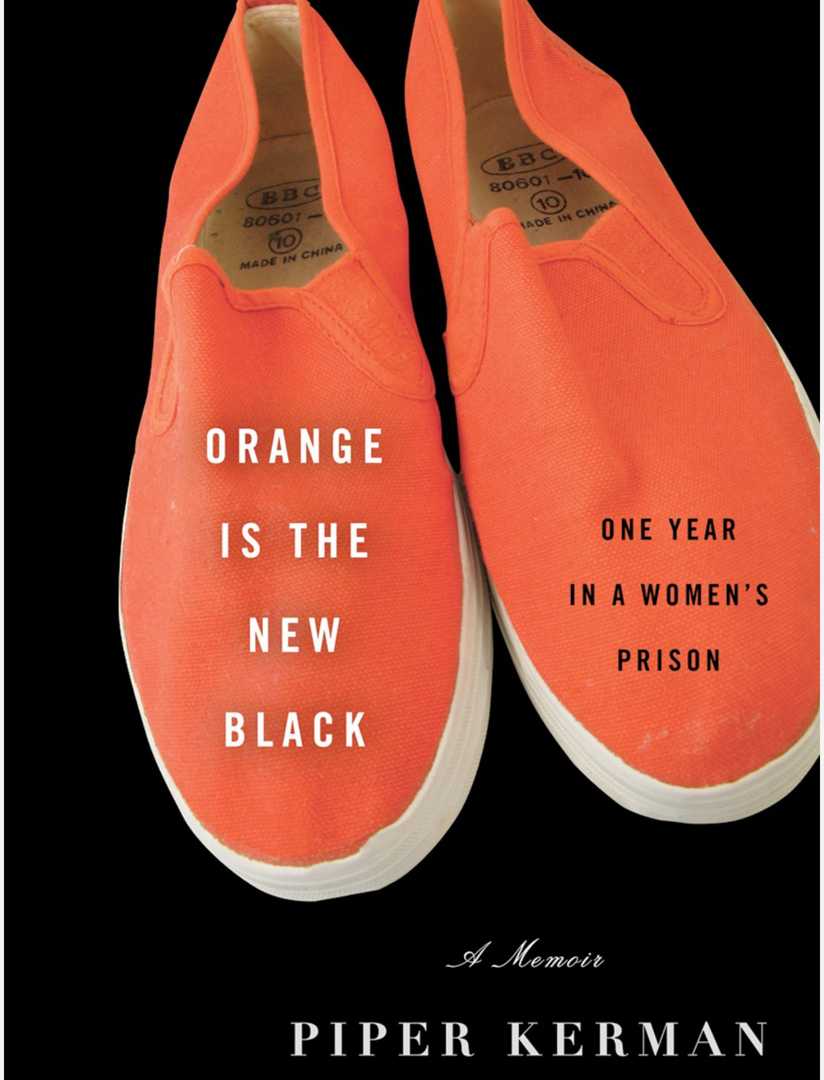
Or too slow.

But hardly any sane, safe drivers like yourself.

Once you see it...

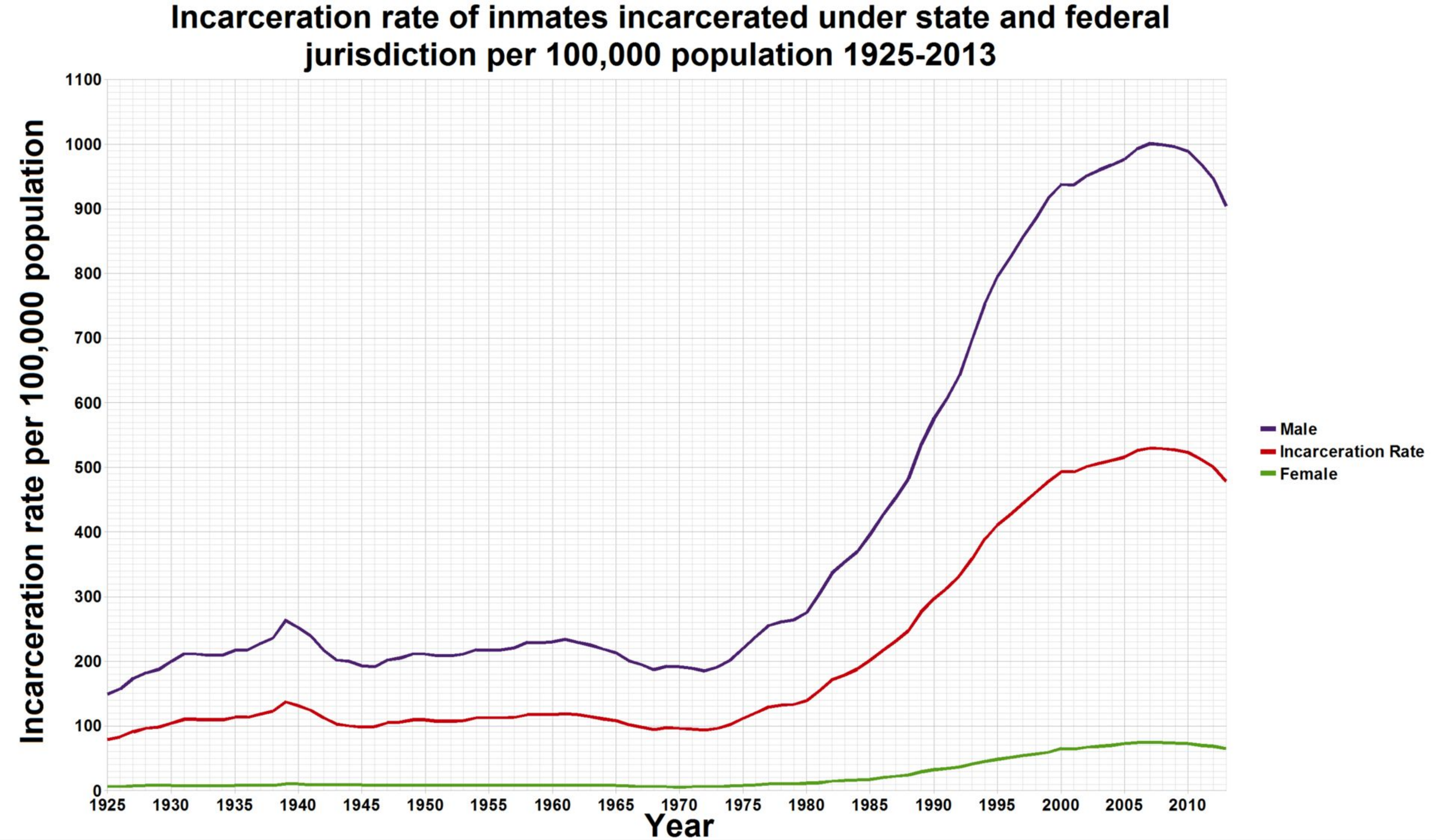
Once you see it...

You see it everywhere.



Long sentences?

Kerman expresses surprise
at the long sentences
her fellow prisoners are serving.



https://commons.wikimedia.org/wiki/File:U.S._incarceration_rates_1925_onwards.png

But also...

Arrive at a random time.

Choose a random prisoner.

Prisoner with sentence x is oversampled by x.

A-Z Topics • Site Map • FOIA

Search bop.gov

Federal Bureau of Prisons

Correctional Excellence. Respect. Integrity.

Home | About Us | Inmates | Locations | Jobs | Business | Resources | Contact Us

Statistics

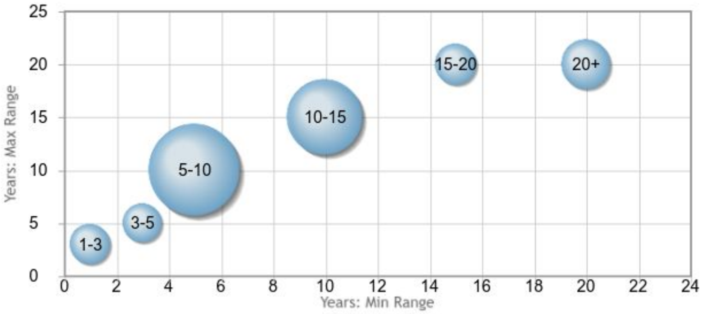
Inmate Statistics | Population Statistics | Staff Statistics

Inmate Statistics

- Age
- Citizenship
- Ethnicity
- Gender
- Offenses
- Prison Safety
- Prison Security Levels
- Programs
- Race
- Release Numbers
- Restricted Housing
- **Sentences Imposed**

Sentences Imposed

Statistics based on prior month's data -- -- Last Updated: Saturday, 26 October 2019
Please Note: Data is limited by availability of sentencing information for inmates in BOP custody.



Sentence	# of Inmates	% of Inmates
0 to 1 year*	4,929	2.3 %
> 1 year to < 3 years**	18,726	11.4%
3 years to < 5 years	17,811	10.8%
5 years to < 10 years	41,987	25.6%
10 years to < 15 years	34,917	21.2%
15 years to < 20 years	18,710	11.4%
20 years or more but < Life	22,655	13.8%
Life	4,536	2.8%

* *The sentence category "0 to 1 year" includes misdemeanor offenses (0-12 months).*
** *The sentence category "> 1 to < 3 years" includes the common sentence type: "Twelve months plus 1 day."*

https://www.bop.gov/about/statistics/statistics_inmate_sentences.jsp

Read the fine print...

This is based on a sample of current prisoners.

So the sample is length-biased.

But we can unbias it!

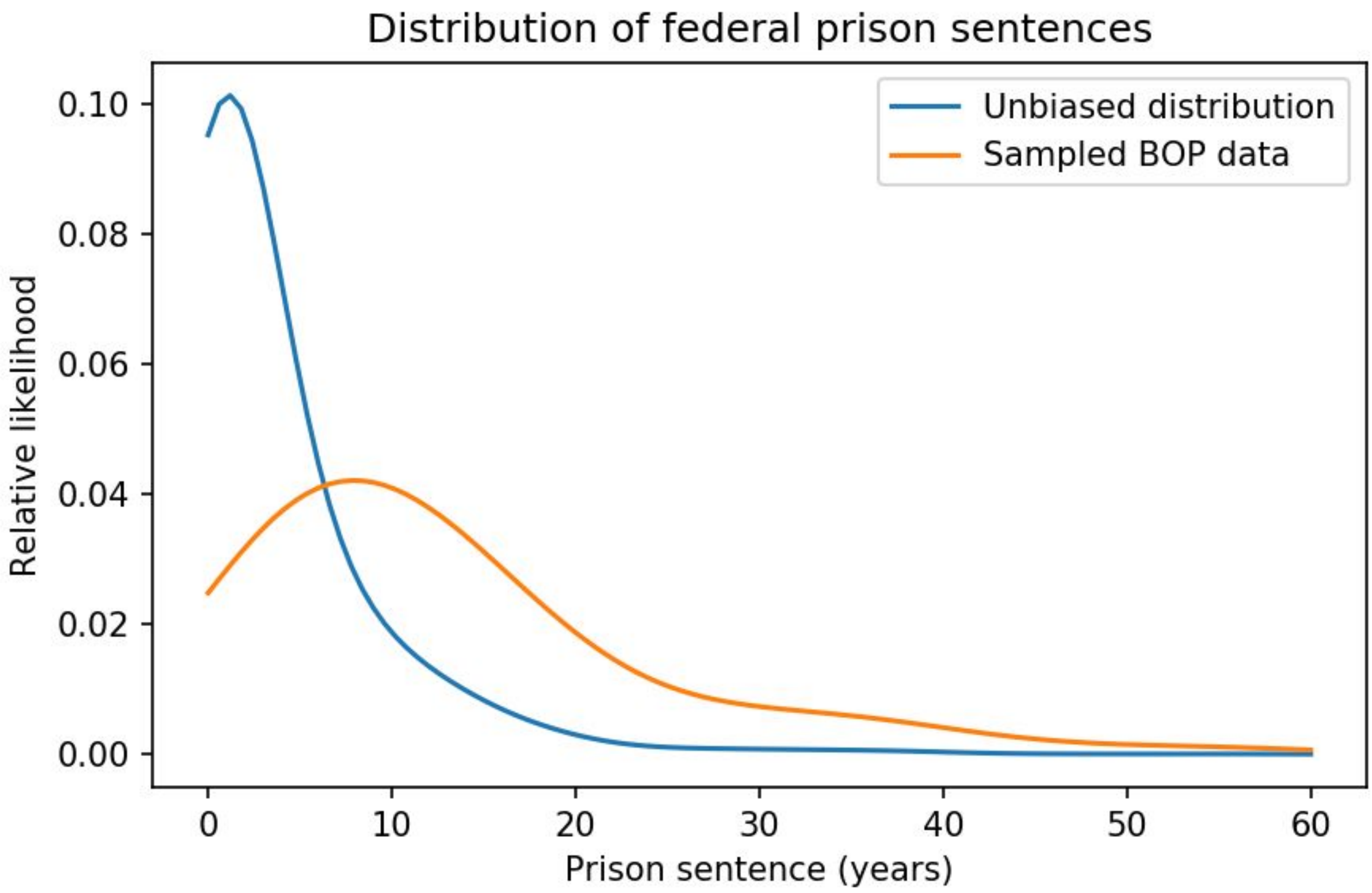
Prisoners in federal prison typically serve 85% of their nominal sentence.

We can take that into account in the weights.

```
weights = 1 / (0.85 * np.array(biased))
```

Here's the unbiased sample.

```
unbiased = resample_weighted(biased, weights)
```


Distribution of federal prison sentences
Unbiased distribution
Sampled BOP data
Relative likelihood
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
Prison sentence (years)

But it depends on how long you stay

Single observation: biased by x.

Observe for a long period: unbiased.

What if you observe for 13 months?

Interval inspection

If the inspection interval is `t`
a prisoner with sentence `x`
is oversampled by `x + t`.

Small `t`: converges to `x`.

Large `t`: converges to constant (unbiased).

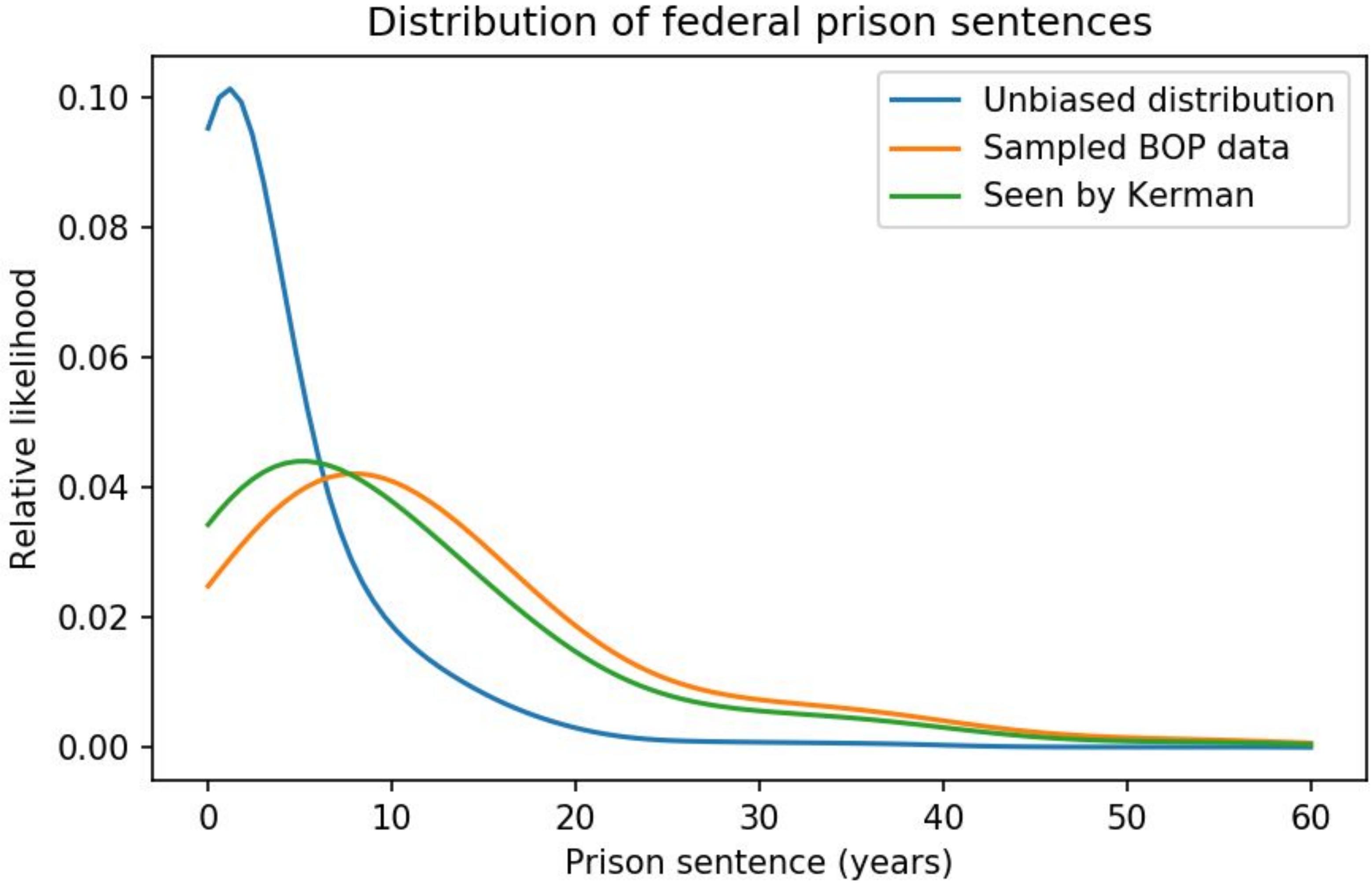
We can also compute the distribution of sentences as seen by someone at the prison for 13 months.

```
x = 0.85 * unbiased
t = 13 / 12

weights = x + t
```

Here's the sample.

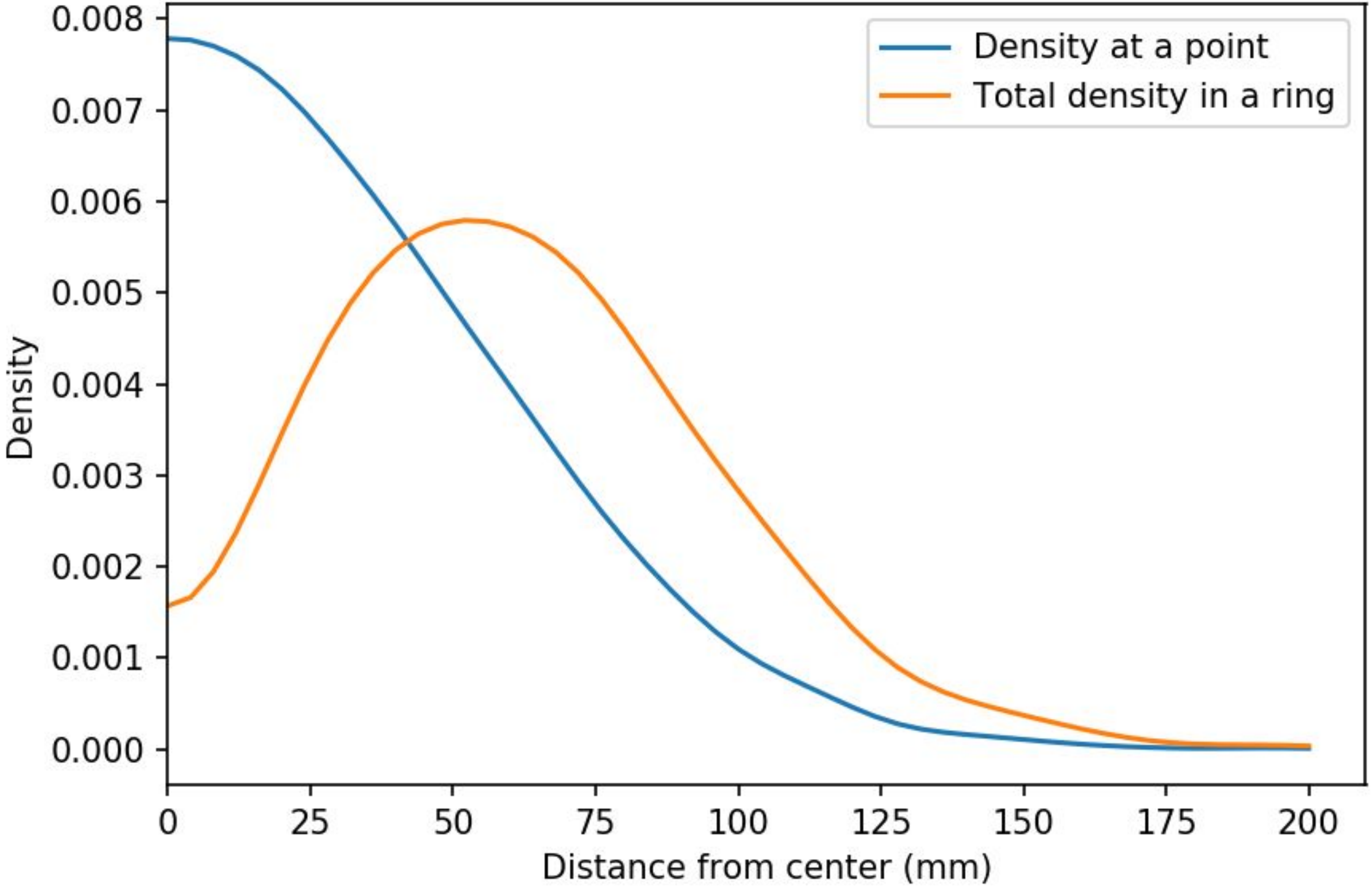
```
kerman = resample_weighted(unbiased, weights)
```


Distribution of federal prison sentences
Unbiased distribution
Sampled BOP data
Seen by Kerman
Relative likelihood
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
Prison sentence (years)

The dartboard paradox

This happens in multiple dimensions, too.

Even more so.



Density at a point
Total density in a ring
Density
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
25
50
75
100
125
150
175
200
Distance from center (mm)

Details in the notebook.

And on my blog,
Probably Overthinking It.

allendowney.com/blog

THE DARTBOARD PARADOX

October 18, 2019 AllenDowney

On November 5, 2019, I will be at PyData NYC to give a talk called The Inspection Paradox is Everywhere. Here's the abstract:

> *The inspection paradox is a statistical illusion you've probably never*

ABOUT ME

Allen Downey is a professor at Olin College and the author of *Think Python*, *Think Bayes*, and other books available from Green Tea Press.

Summary

Length-biased sampling is everywhere.

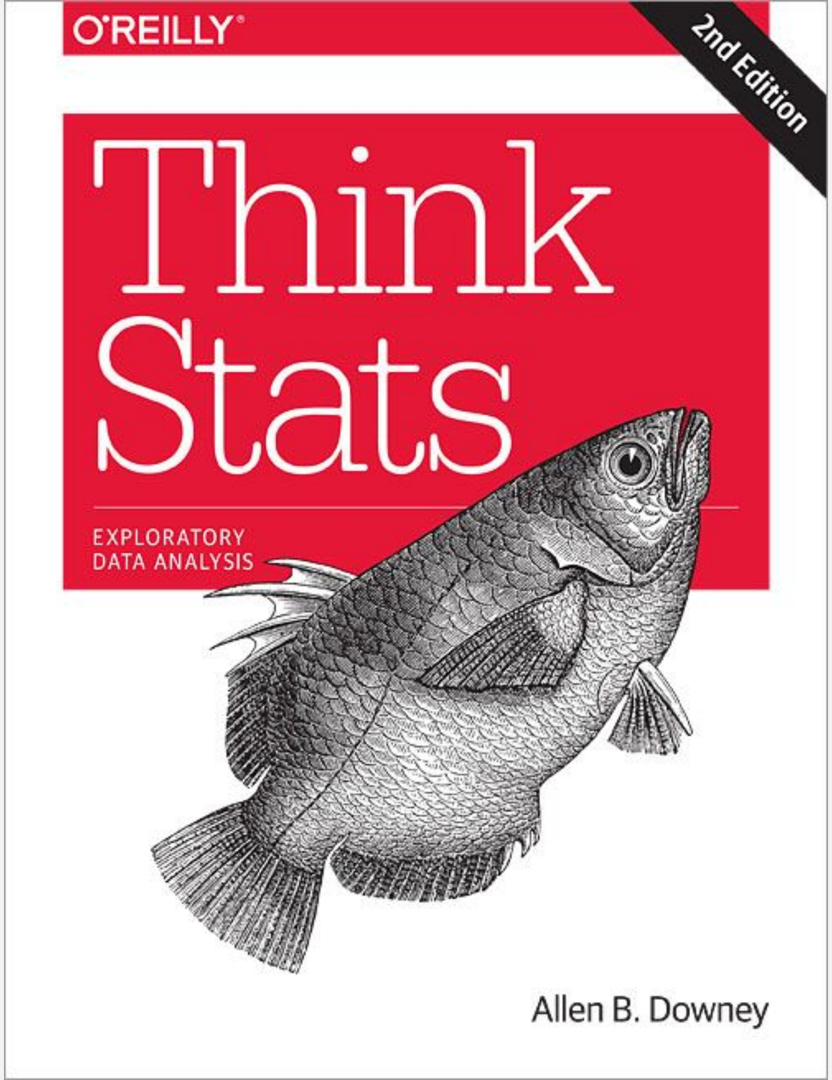
Can be a subtle source of error.

Also an opportunity for clever experimental design.

More reading

Class size example:

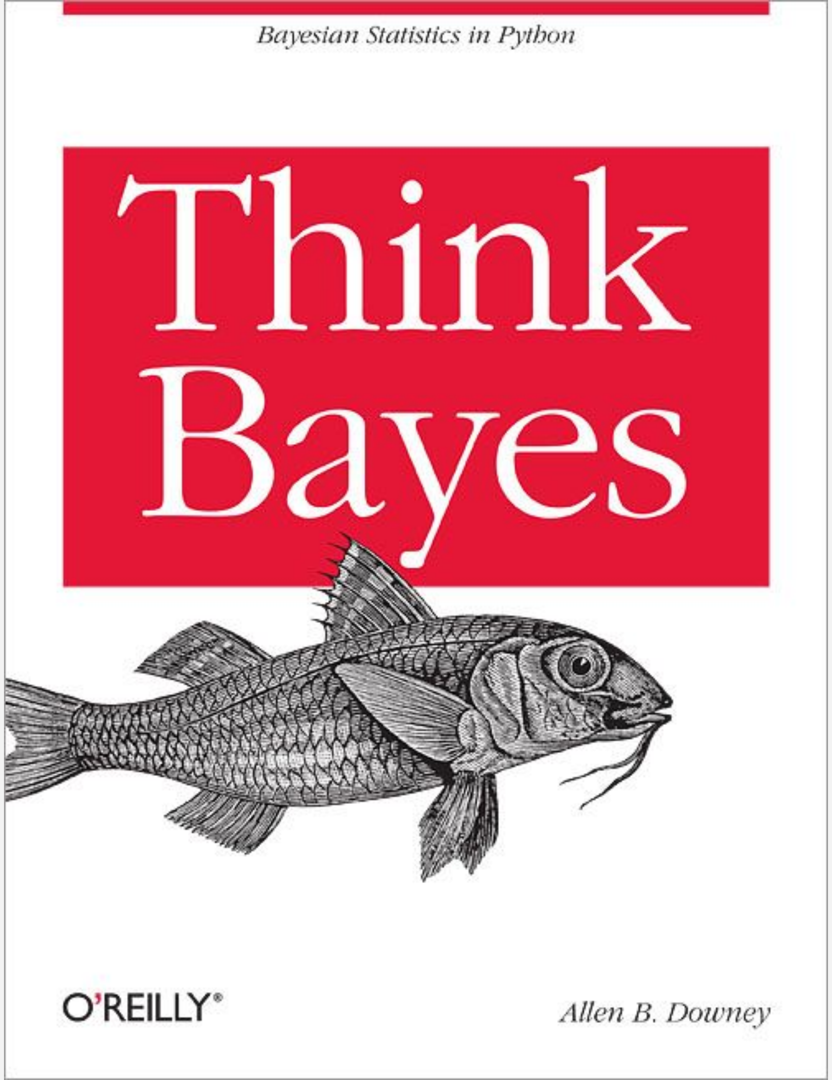
Think Stats, Chapter 3



More reading

Red line example:

Think Bayes, Chapter 8



Towards Data Science

DATA SCIENCE MACHINE LEARNING PROGRAMMING VISUALIZATION AI JOURNALISM MORE | CONTRIBUTE

Upgrade

The Inspection Paradox is Everywhere

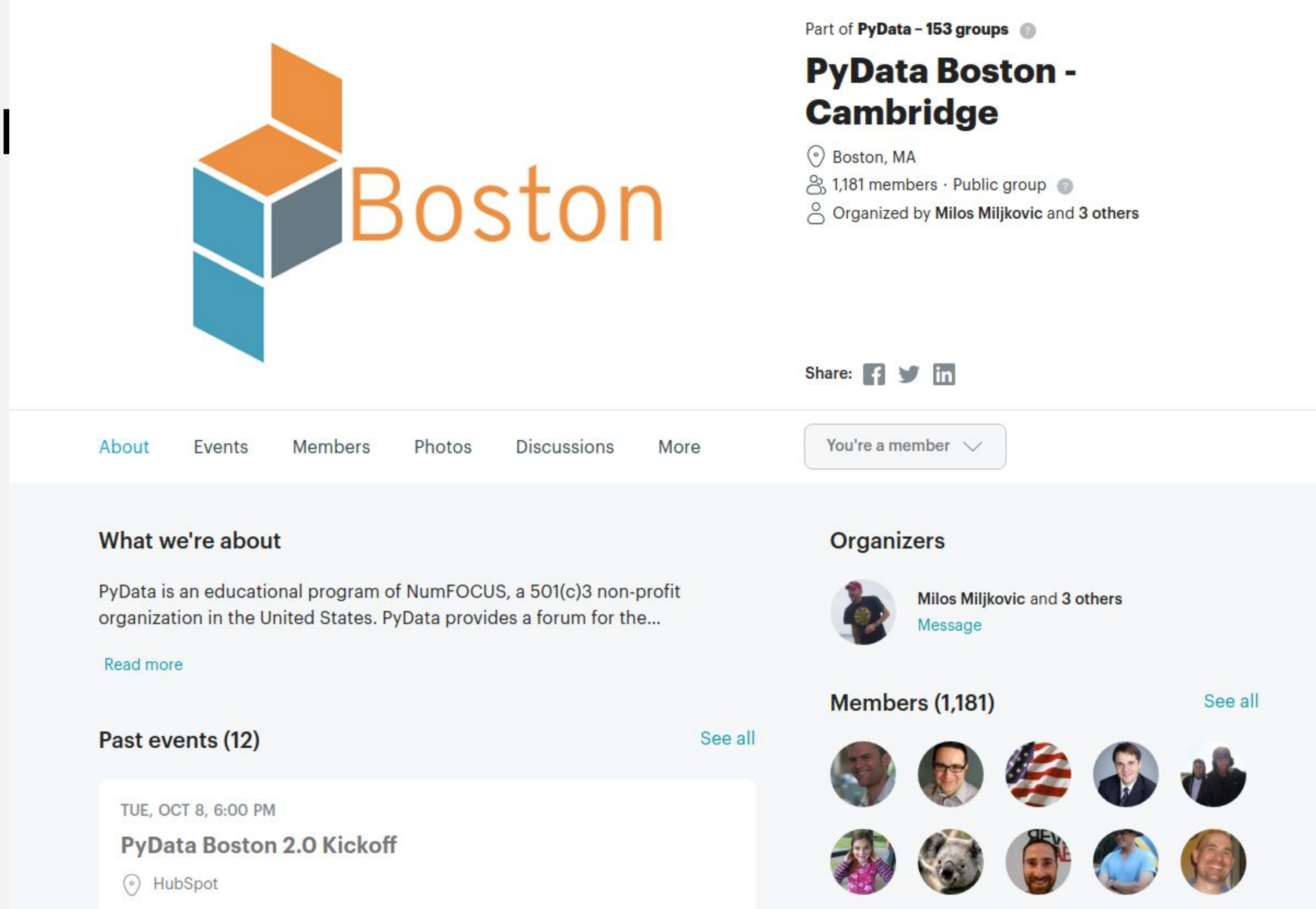
Allen Downey
Aug 5 · 8 min read ★

The inspection paradox is a statistical illusion you've probably never heard of. It's a common source of confusion, an occasional cause of error, and an opportunity for clever experimental design.

And once you know about it, you see it everywhere.

https://towardsdatascience.com/the-inspection-paradox-is-everywhere-2ef1c2e9d709?source=friends_link&sk=a38a2925e44f481d27e5b2452128716a

Pyl



https://www.meetup.com/PyData-Boston-Cambridge/

Help me plan my sabbatical.

WHAT SHOULD I DO?

September 19, 2019 AllenDowney

I am planning to be on sabbatical from June 2020 to August 2021, so I am thinking about how to spend it. Let me tell you what I *can* do, and you can tell me what I *should* do.

Data Science

I consider myself a data scientist, but that means different things to different people. More specifically, I can contribute in the following areas:

- Data exploration, modeling, and prediction,
- Bayesian statistics and machine learning,
- Scientific computing and optimization,
- Software engineering and reproducible science
- Technical communication, including data visualization.

I have written a series of books related to data science and scientific computing, including *Think Stats*, *Think Bayes*, *Physical Modeling in MATLAB*, and *Modeling and Simulation in Python*.

blog

website

github

downey@allendowney.com/blog

twitter

email

Why Your Friends Have More Friends than You Do[1]

Scott L. Feld
State University of New York at Stony Brook