Numerical Summaries

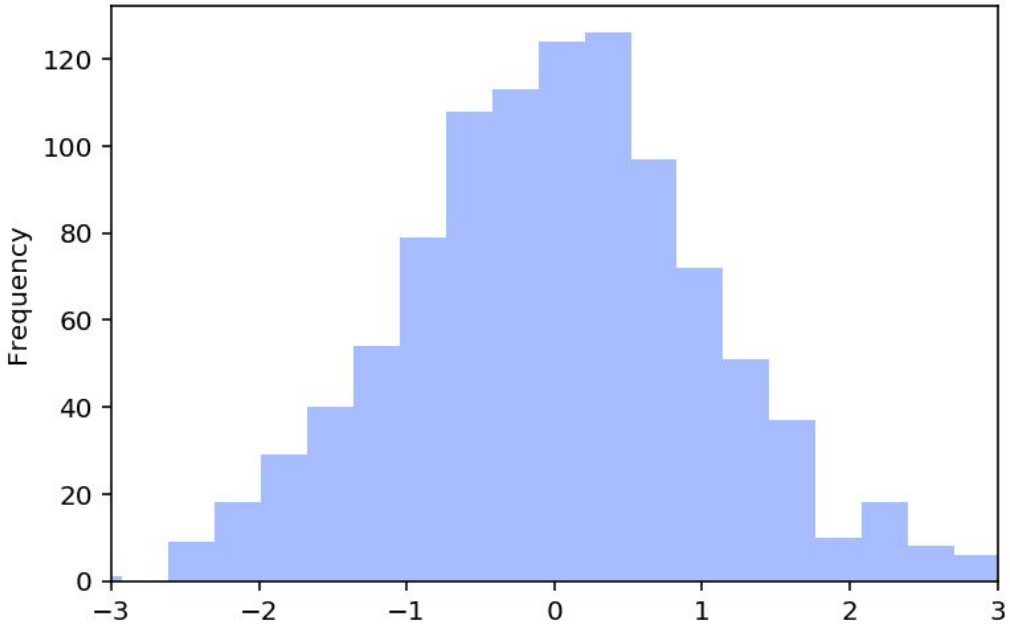
Concept Module 2

Template by: Laurent Lessard

Numerical summaries

We saw qualitative descriptions such as “bimodal” or “skewed”.

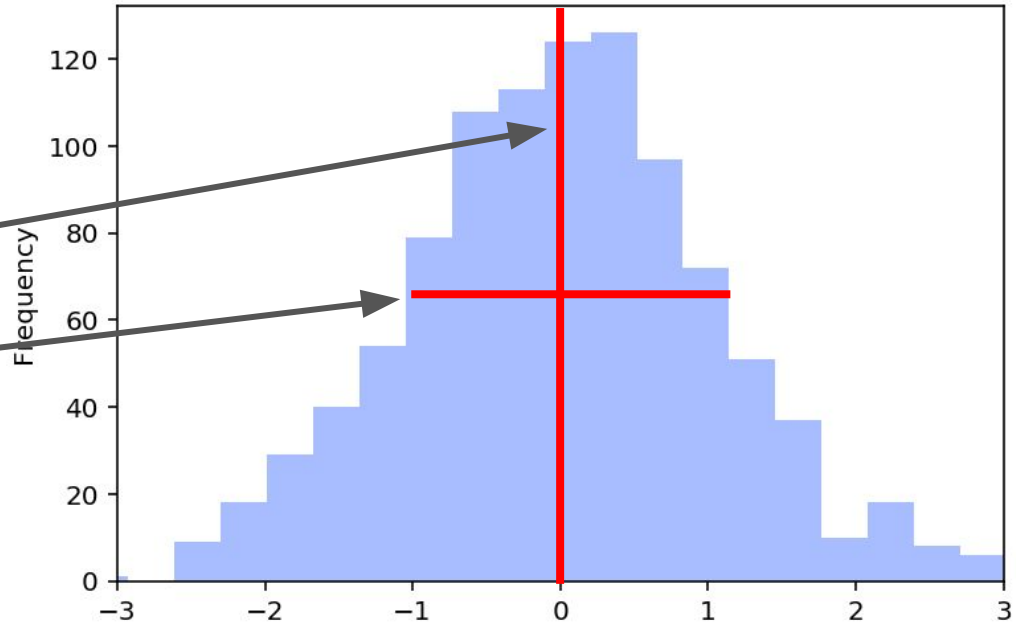
What about quantitative descriptions?



Numerical summaries

We'll focus on quantifying two notions:

- **Location**
- **Spread**



What could the `location` parameter be?

- **Goal:** represent a "typical" value of the data.
- One way: use the average (also called the mean)

- If data are in a list `x` of length N, the mean `m` is:

$$m = \frac{x[0] + x[1] + \ldots + x[N-1]}{N}$$

Calculating the mean

```
df.flip_value.mean()
```

```
0.4958
```

```
df.mean()
```

```
flip_value            0.4958
number_of_flips    5000.5000
dtype: float64
```

	flip_value	number_of_flips
0	0	1
1	0	2
2	0	3
3	0	4
4	1	5

Results from flipping a coin 10,000 times.
Heads = 1, Tails = 0.

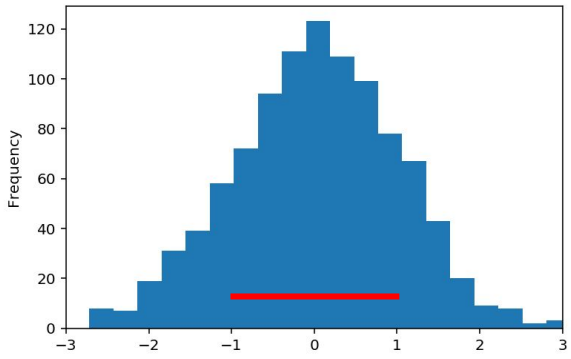
0.4958 is the fraction of the time the flip came up Heads. The mean is not necessarily achievable. Coin flips can only produce Heads (1) or Tails (0) !

What could the `spread` parameter be?

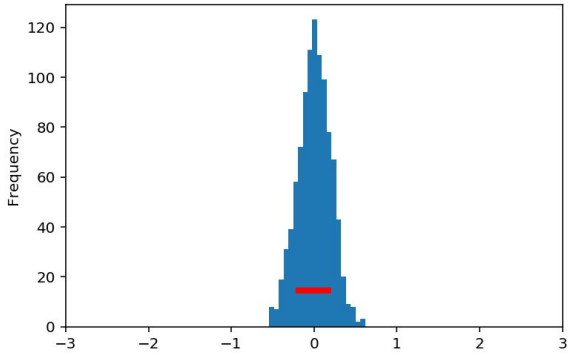
- **Goal:** represent amount of deviation from the mean. One way is by using the notion of standard deviation.

$$s = \sqrt{\frac{(x[0]-m)^{**}2 + (x[1]-m)^{**}2 + \ldots + (x[N-1]-m)^{**}2}{N-1}}$$

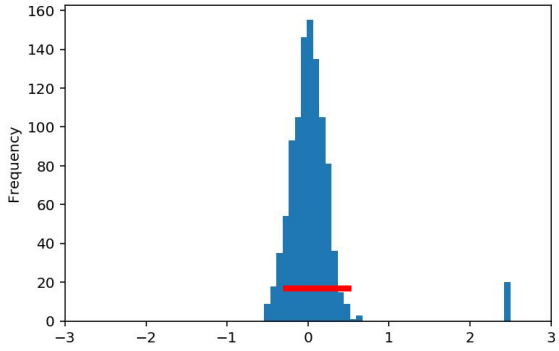
s = 1.0043



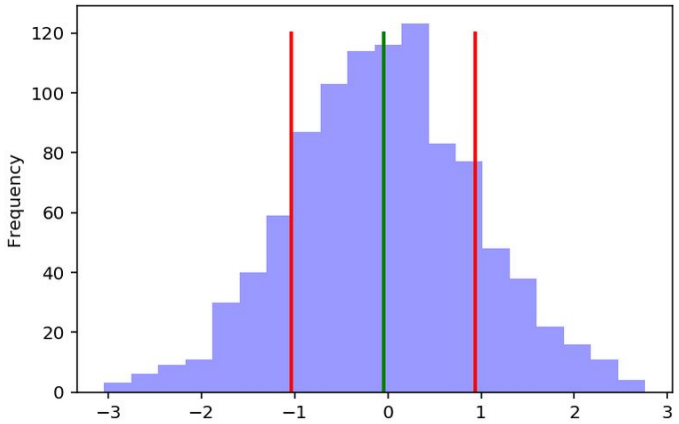
0.19850

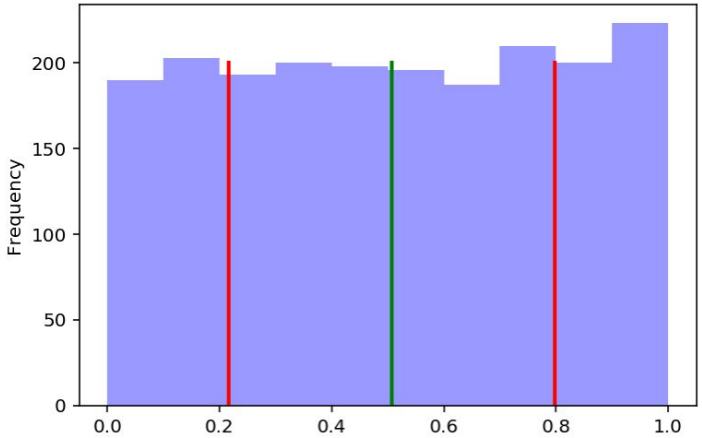


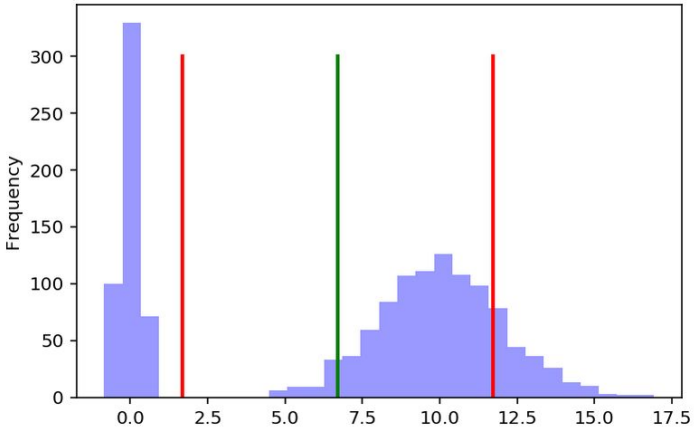
0.39683



Example distributions







Green lines: mean

Red lines: mean ± one standard deviation

Calculation of standard deviation

```
df.flip_value.std()
```

```
0.39683511413531947
```

```
df.std()
```

```
flip_value            0.396835
number_of_flips    2886.895680
dtype: float64
```

	flip_value	number_of_flips
0	0	1
1	0	2
2	0	3
3	0	4
4	1	5

Results from flipping a coin 10,000 times.
Heads = 1, Tails = 0.

Manual calculation

```
# this is the same as:  m = df.mean()

m = df.sum() / df.count()
```

```
# this is the same as:  s = df.std()

import numpy as np

m = df.sum() / df.count()
s = np.sqrt( ((df-m)**2).sum() / (df.count()-1) )
```

WARNING: Beware of summaries

Mean and standard deviation are only two numbers!

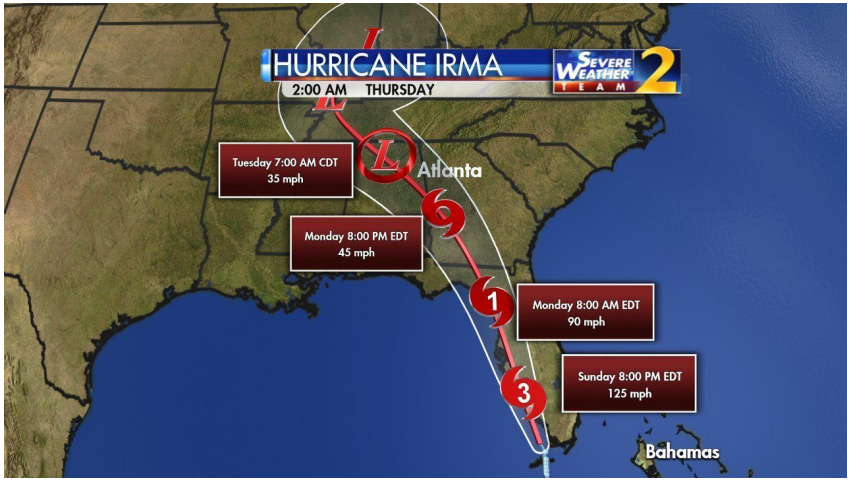
- Typically cannot convey the same information as the entire dataset.
- Sometimes, they can convey misleading information.

When can using the mean cause problems?

Two examples:

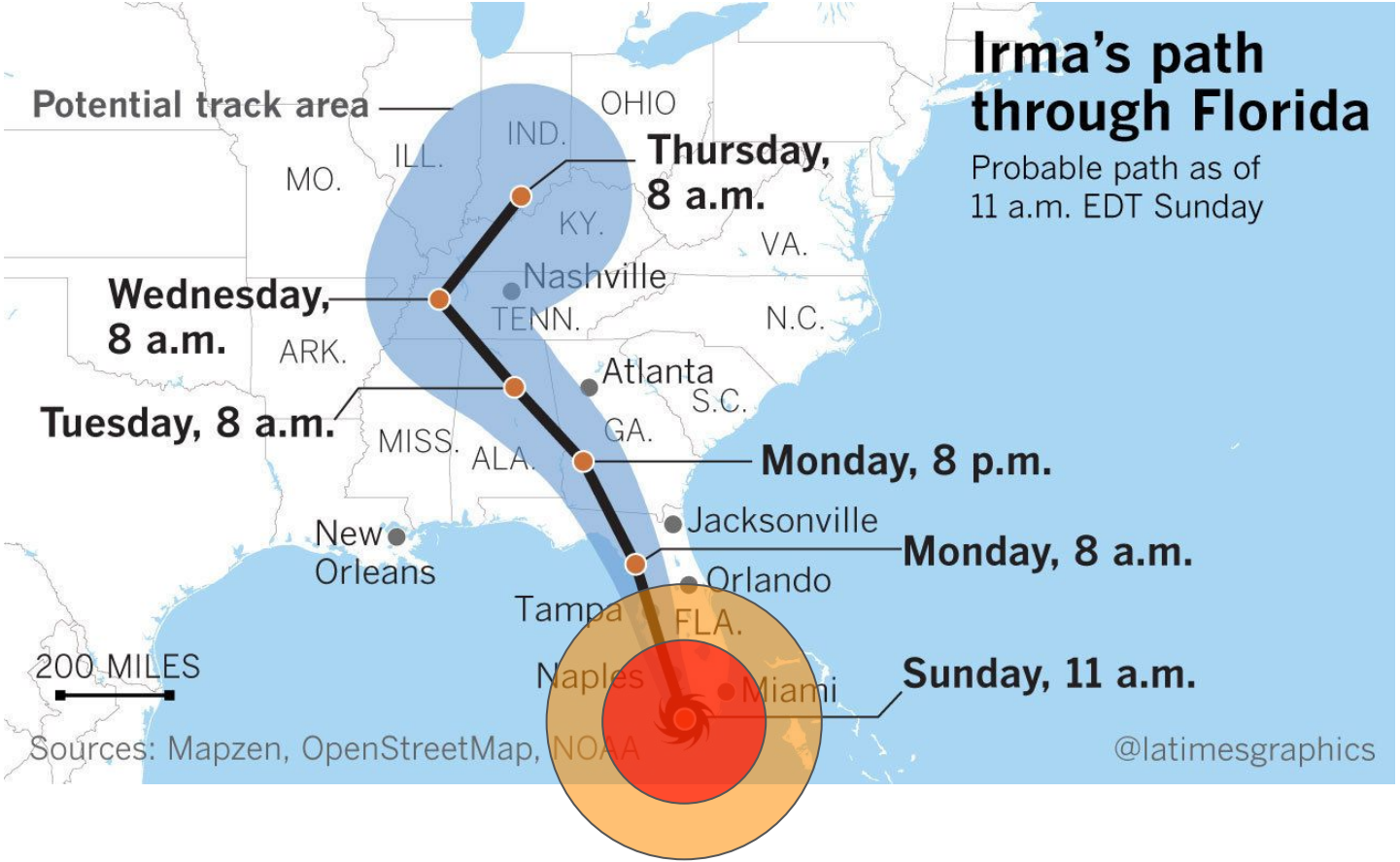
1. Mean estimation
2. Outliers

Mean estimation



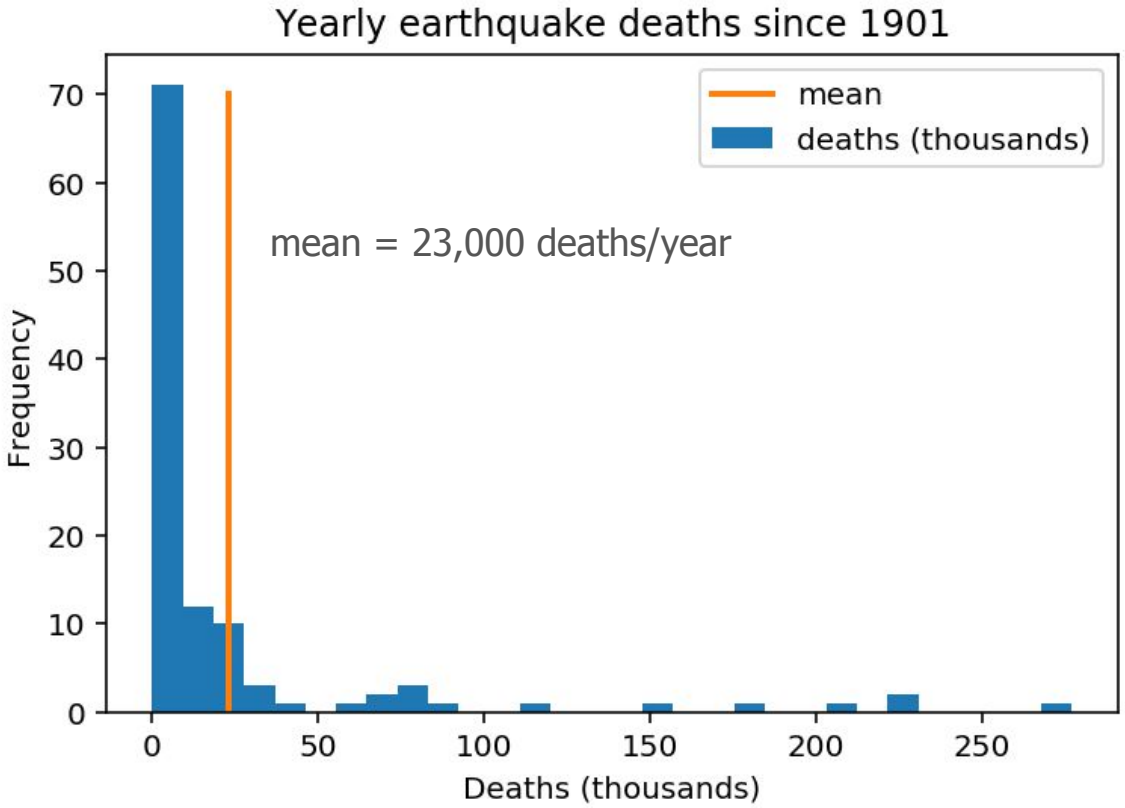
Hurricane Irma (Sept. 2017)

Mean estimation



The shaded blue area represents where the *center* of the hurricane could possibly be, not the *width* of the hurricane.

Outliers

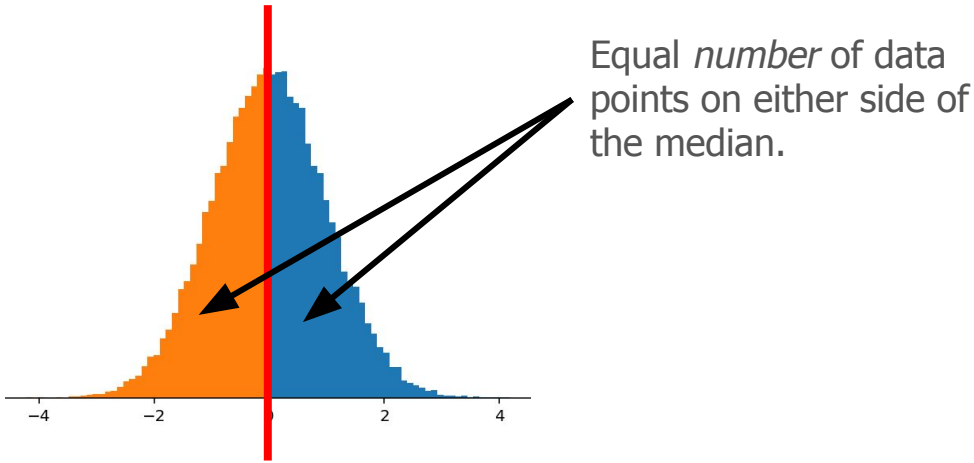


The mean is not a "typical" value

A small number of "outliers" or massive earthquakes inflate the mean.

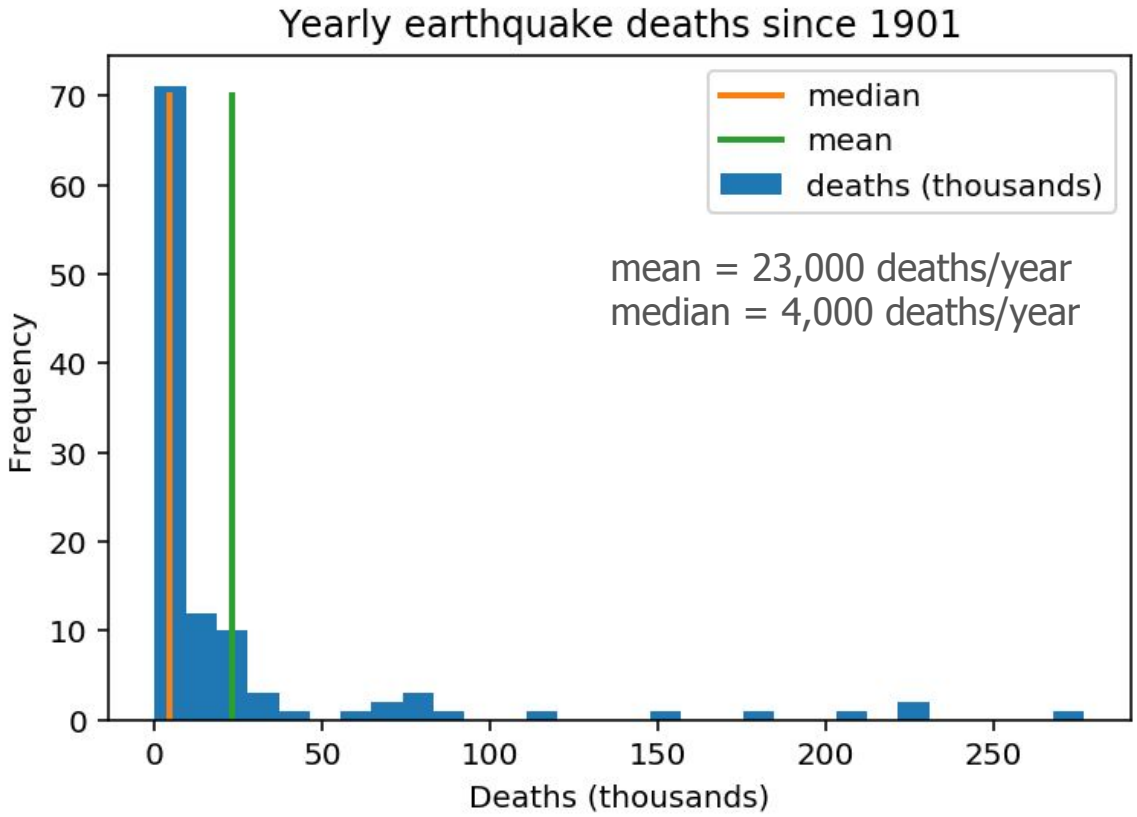
Median

- One `location` parameter robust to outliers: the median
- Separates lower half of data from upper half

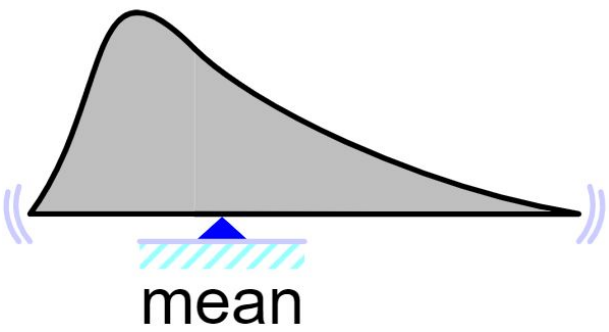


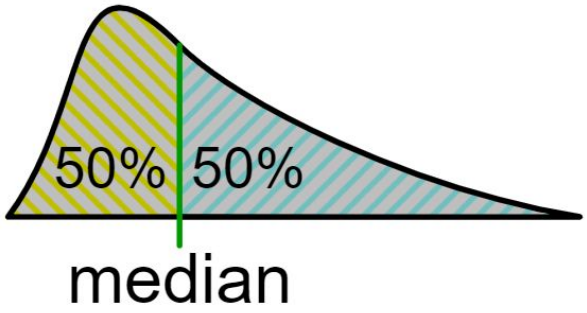
https://en.wikipedia.org/wiki/Median

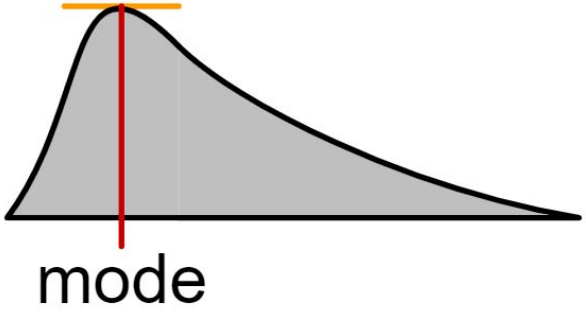
Median


Yearly earthquake deaths since 1901
median
mean
deaths (thousands)
mean = 23,000 deaths/year
median = 4,000 deaths/year
Frequency
0
10
20
30
40
50
60
70
0
50
100
150
200
250
Deaths (thousands)

Measures of Central Tendency







Mean is the arithmetic average of a list of numbers (also the center of mass!)

```
df.mean()
```

Median is the middle value in a sorted list of numbers (half of the area on each side)

```
df.median()
```

Mode is the most frequently occuring value in a list of numbers (the peak)

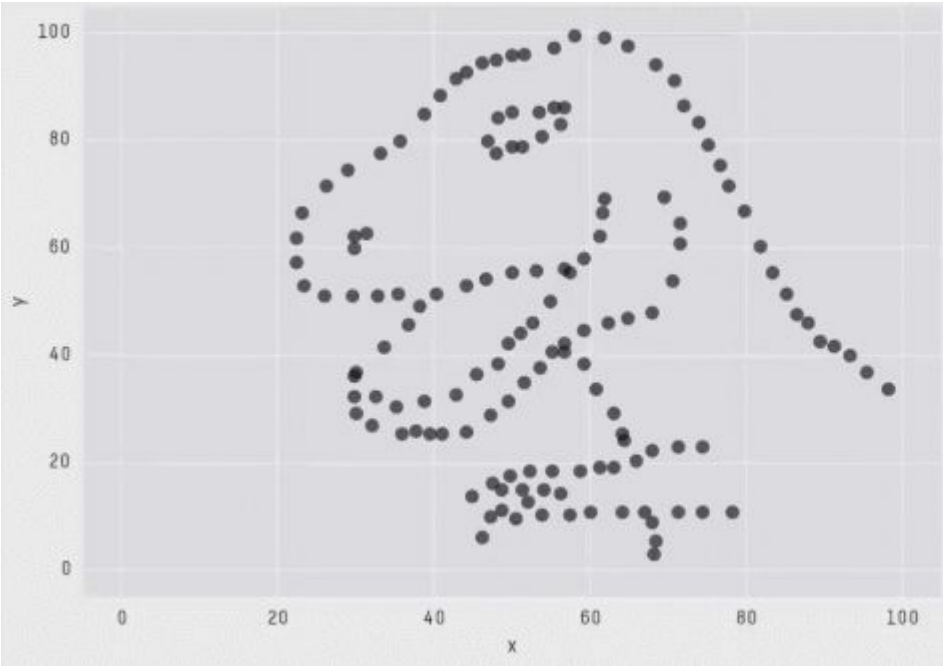
```
df.mode()
```

Summary

- Numerical summaries can be descriptive and useful.
- They can also be misleading.
- Always graph your data first!

Dinosaurus data set:

Each 2D scatter plot (and all intermediate plots) below have the same mean and standard deviation in both x and y directions!



X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

Source: https://www.autodeskresearch.com/publications/samestats