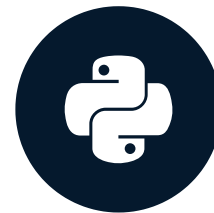# What is statistics?

INTRODUCTION TO STATISTICS IN PYTHON

Maggie Matsui
Content Developer, DataCamp

datacamp

What is statistics?

- **The field of statistics** - the practice and study of collecting and analyzing data

- **A summary statistic** - a fact about or summary of some data

What can statistics do?

What is statistics?

- **The field of statistics** - the practice and study of collecting and analyzing data

- **A summary statistic** - a fact about or summary of some data

What can statistics do?

- How likely is someone to purchase a product? Are people more likely to purchase it if they can use a different payment system?

- How many occupants will your hotel have? How can you optimize occupancy?

- How many sizes of jeans need to be manufactured so they can fit 95% of the population? Should the same number of each size be produced?

- A/B tests: Which ad is more effective in getting people to purchase a product?

What can't statistics do?

- *Why* is *Game of Thrones* so popular?

Instead...

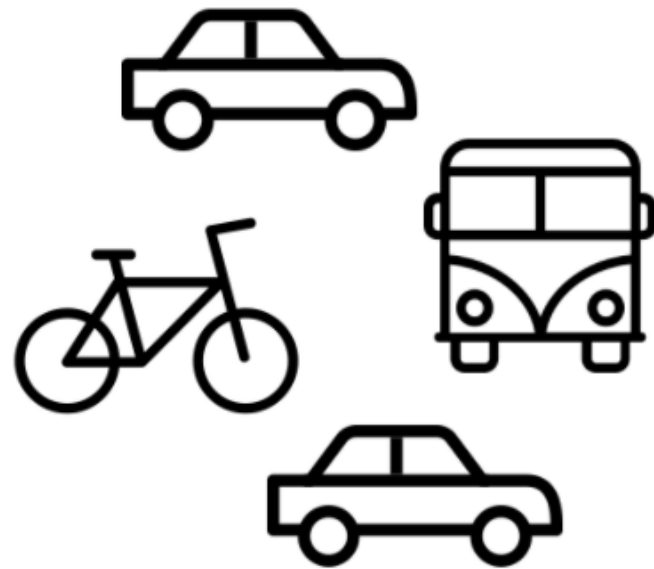- Are series with more violent scenes viewed by more people?

But...

- Even so, this can't tell us if more violent scenes lead to more views
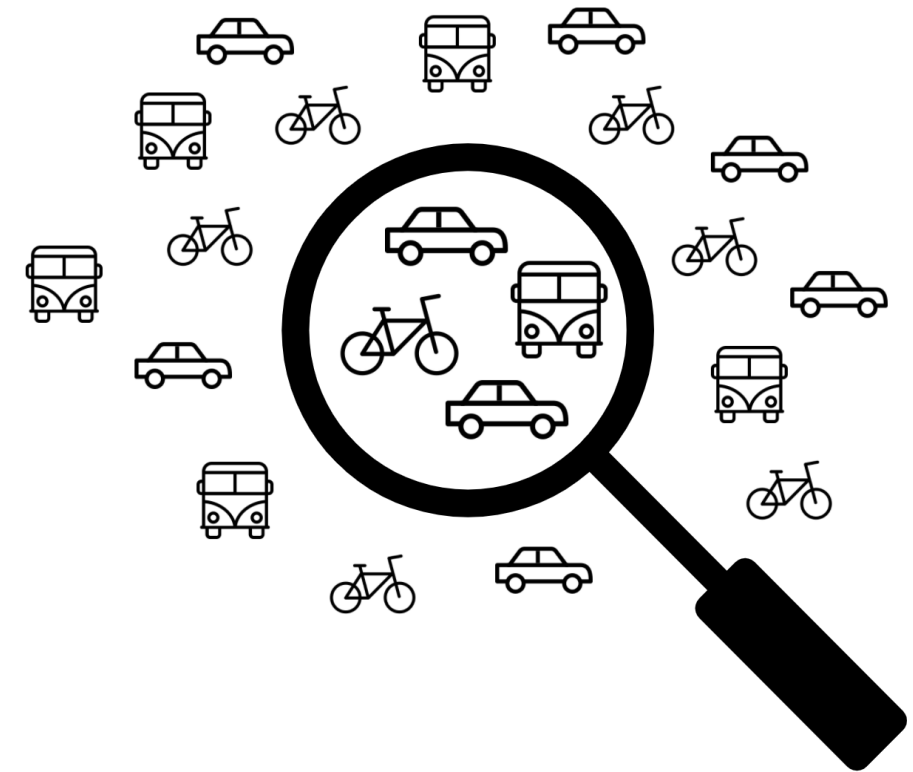
Types of statistics

Descriptive statistics

- *Describe* and summarize data

- 50% of friends drive to work

- 25% take the bus

- 25% bike

Inferential statistics

- Use a sample of data to make *inferences* about a larger population

What percent of people drive to work?

Types of data

Numeric (Quantitative)

- **Continuous (Measured)**
  - Airplane speed
  - Time spent waiting in line
- **Discrete (Counted)**
  - Number of pets
  - Number of packages shipped

Categorical (Qualitative)

- **Nominal (Unordered)**
  - Married/unmarried
  - Country of residence
- **Ordinal (Ordered)**

  - ○ Strongly disagree
  - ○ Somewhat disagree
  - ○ Neither agree nor disagree
  - ◉ Somewhat agree
  - ○ Strongly agree

Categorical data can be represented as numbers

Nominal (Unordered)

- Married/unmarried (`1` / `0`)

- Country of residence (`1` , `2` , ...)

Ordinal (Ordered)

- Strongly disagree (`1`)

- Somewhat disagree (`2`)

- Neither agree nor disagree (`3`)

- Somewhat agree (`4`)
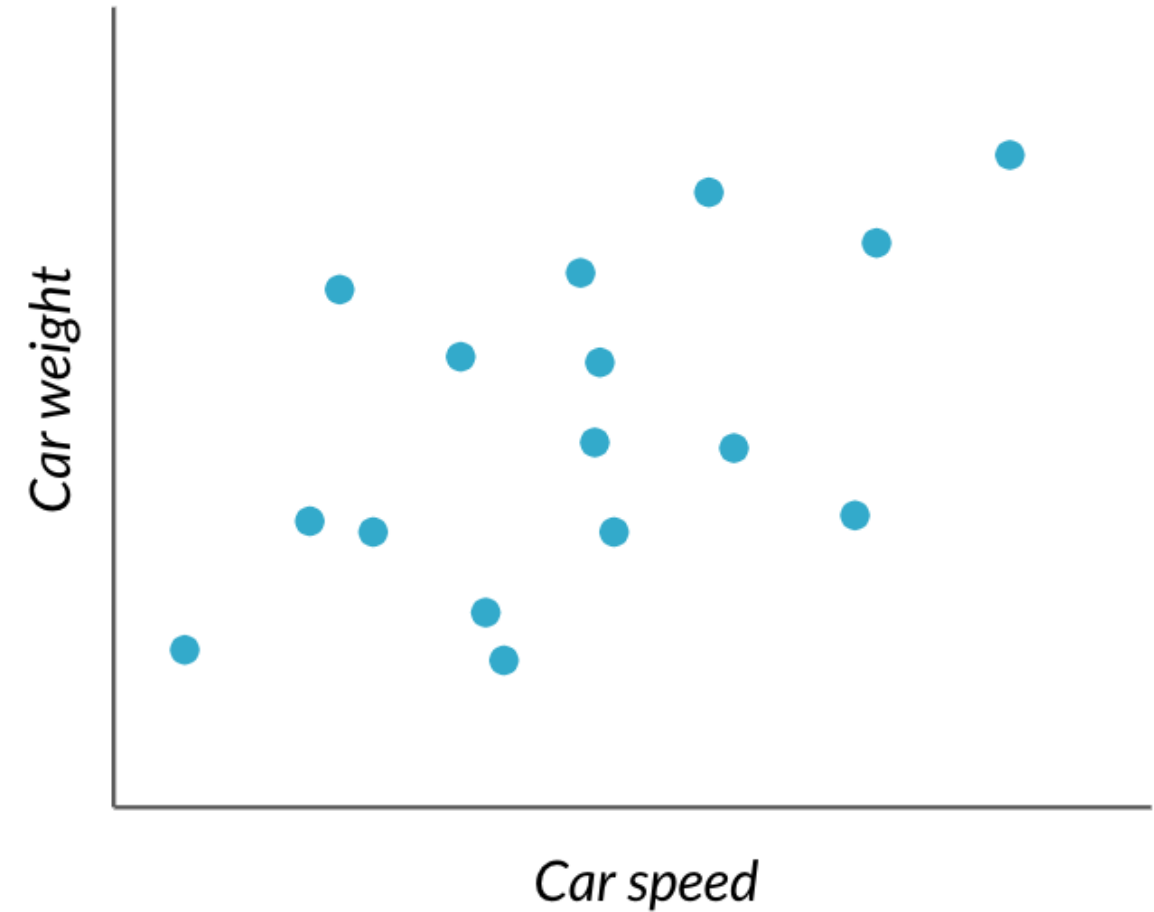
- Strongly agree (`5`)

Why does data type matter?

Summary statistics

```python
import numpy as np

np.mean(car_speeds['speed_mph'])
```
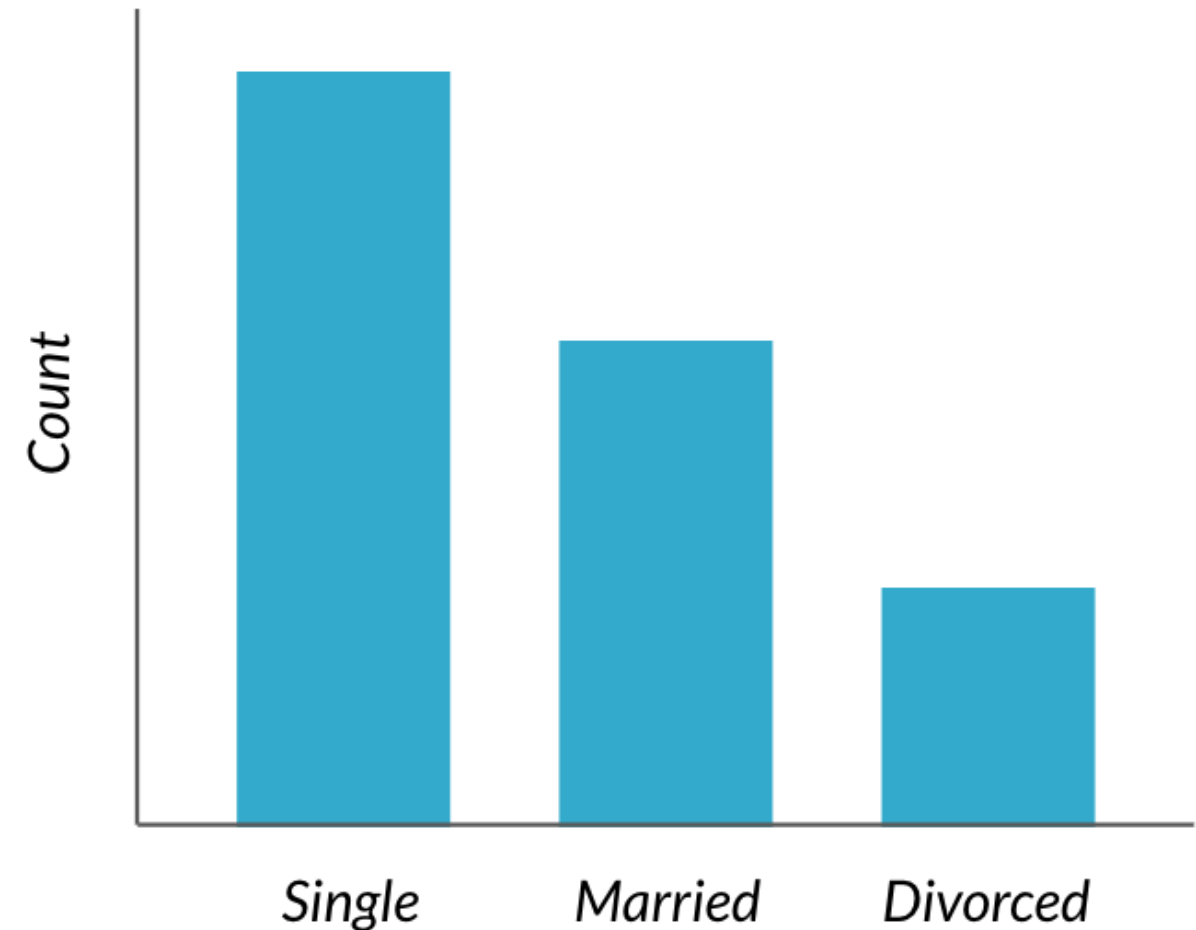
```
40.09062
```

Plots

Why does data type matter?

Summary statistics

```
demographics['marriage_status'].value_counts()
```

```
single      188
married     143
divorced    124
dtype: int64
```
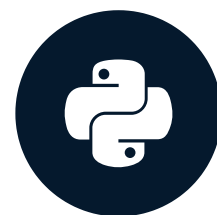
Plots

Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON

Measures of center

INTRODUCTION TO STATISTICS IN PYTHON

**Maggie Matsui**
Content Developer, DataCamp

Mammal sleep data

```
print(msleep)
```

	name	genus	vore	order	...	sleep_cycle	awake	brainwt	bodywt
1	Cheetah	Acinonyx	carni	Carnivora	...	NaN	11.9	NaN	50.000
2	Owl monkey	Aotus	omni	Primates	...	NaN	7.0	0.01550	0.480
3	Mountain beaver	Aplodontia	herbi	Rodentia	...	NaN	9.6	NaN	1.350
4	Greater short-ta...	Blarina	omni	Soricomorpha	...	0.133333	9.1	0.00029	0.019
5	Cow	Bos	herbi	Artiodactyla	...	0.666667	20.0	0.42300	600.000
..	...	...	...	...	...	...	...	...	...
79	Tree shrew	Tupaia	omni	Scandentia	...	0.233333	15.1	0.00250	0.104
80	Bottle-nosed do...	Tursiops	carni	Cetacea	...	NaN	18.8	NaN	173.330
81	Genet	Genetta	carni	Carnivora	...	NaN	17.7	0.01750	2.000
82	Arctic fox	Vulpes	carni	Carnivora	...	NaN	11.5	0.04450	3.380
83	Red fox	Vulpes	carni	Carnivora	...	0.350000	14.2	0.05040	4.230

Histograms

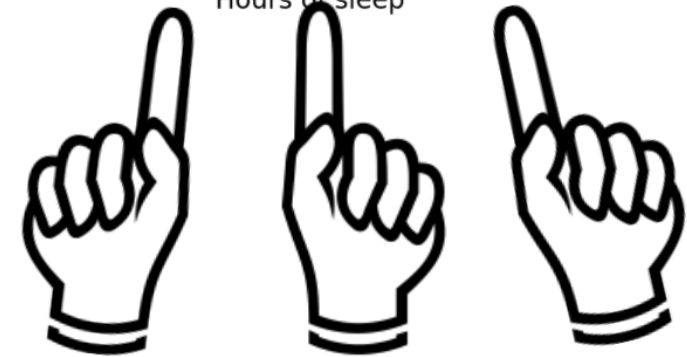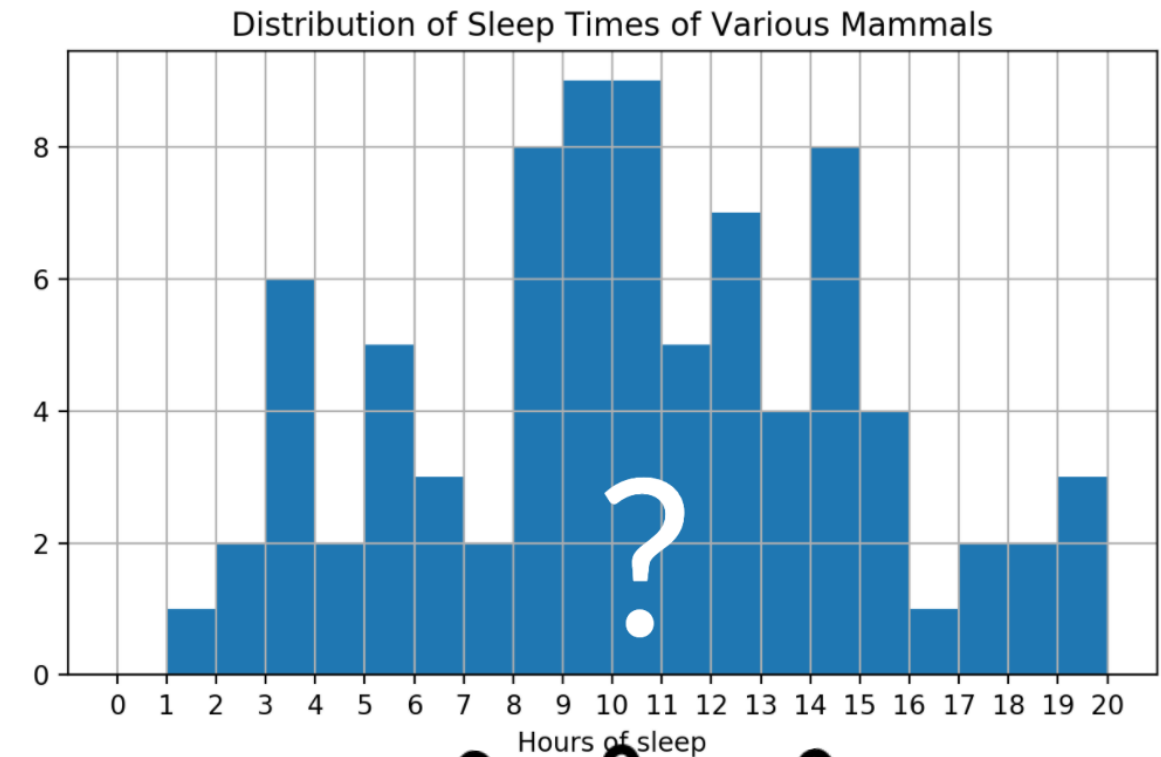Distribution of Sleep Times of Various Mammals
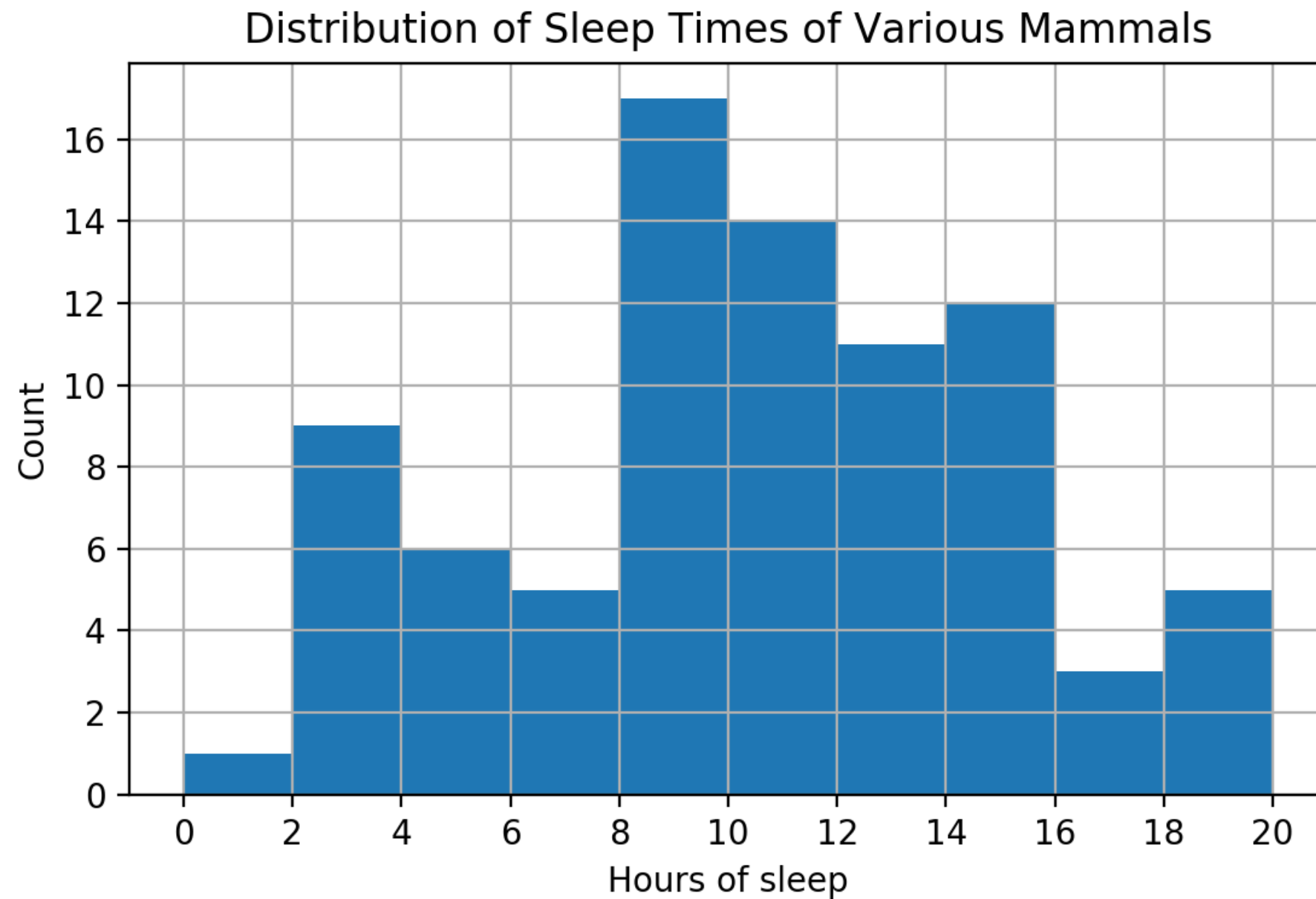
How long do mammals in this dataset typically sleep?

What's a typical value?

Where is the center of the data?

- Mean

- Median

- Mode

Measures of center: mean

	name	sleep_total
1	Cheetah	12.1
2	Owl monkey	17.0
3	Mountain beaver	14.4
4	Greater short-t...	14.9
5	Cow	4.0
..	...	...

```python
import numpy as np
np.mean(msleep['sleep_total'])
```

```
10.43373
```

Mean sleep time $=$

$$\frac{12.1 + 17.0 + 14.4 + 14.9 + ...}{83} = 10.43$$

Measures of center: median

```python
msleep['sleep_total'].sort_values()
```

```
29     1.9
30     2.7
22     2.9
9      3.0
23     3.1
     ...
19    18.0
61    18.1
36    19.4
21    19.7
42    19.9
```

```python
msleep['sleep_total'].sort_values().iloc[41]
```

```
10.1
```

```python
np.median(msleep['sleep_total'])
```

```
10.1
```

Measures of center: mode

Most frequent value

```
msleep['sleep_total'].value_counts()
```

```
12.5    4
10.1    3
14.9    2
11.0    2
8.4     2
...
14.3    1
17.0    1
Name: sleep_total, Length: 65, dtype: int64
```

```
msleep['vore'].value_counts()
```

```
herbi      32
omni       20
carni      19
insecti     5
Name: vore, dtype: int64
```

```
import statistics
statistics.mode(msleep['vore'])
```

```
'herbi'
```

Adding an outlier

```python
# Subset msleep to select rows where 'vore' equals 'insecti'
msleep[msleep['vore'] == 'insecti']
```

```
                    name        genus     vore          order  sleep_total
22         Big brown bat    Eptesicus   insecti     Chiroptera         19.7
43      Little brown bat       Myotis   insecti     Chiroptera         19.9
62        Giant armadillo   Priodontes  insecti      Cingulata         18.1
67  Eastern american mole    Scalopus   insecti   Soricomorpha          8.4
```

Adding an outlier

```
msleep[msleep['vore'] == "insecti"]['sleep_total'].agg([np.mean, np.median])
```

```
mean        16.53
median      18.9
Name: sleep_total, dtype: float64
```

Adding an outlier

```
msleep[msleep['vore'] == 'insecti']
```

```
                    name       genus    vore         order  sleep_total
22             Big brown bat   Eptesicus  insecti    Chiroptera       19.7
43          Little brown bat      Myotis  insecti    Chiroptera       19.9
62            Giant armadillo  Priodontes  insecti     Cingulata       18.1
67       Eastern american mole    Scalopus  insecti  Soricomorpha        8.4
84       Mystery insectivore         ...  insecti          ...        0.0
```

Adding an outlier

```
msleep[msleep['vore'] == "insecti"]['sleep_total'].agg([np.mean, np.median])
```

```
mean        13.22
median      18.1
Name: sleep_total, dtype: float64
```
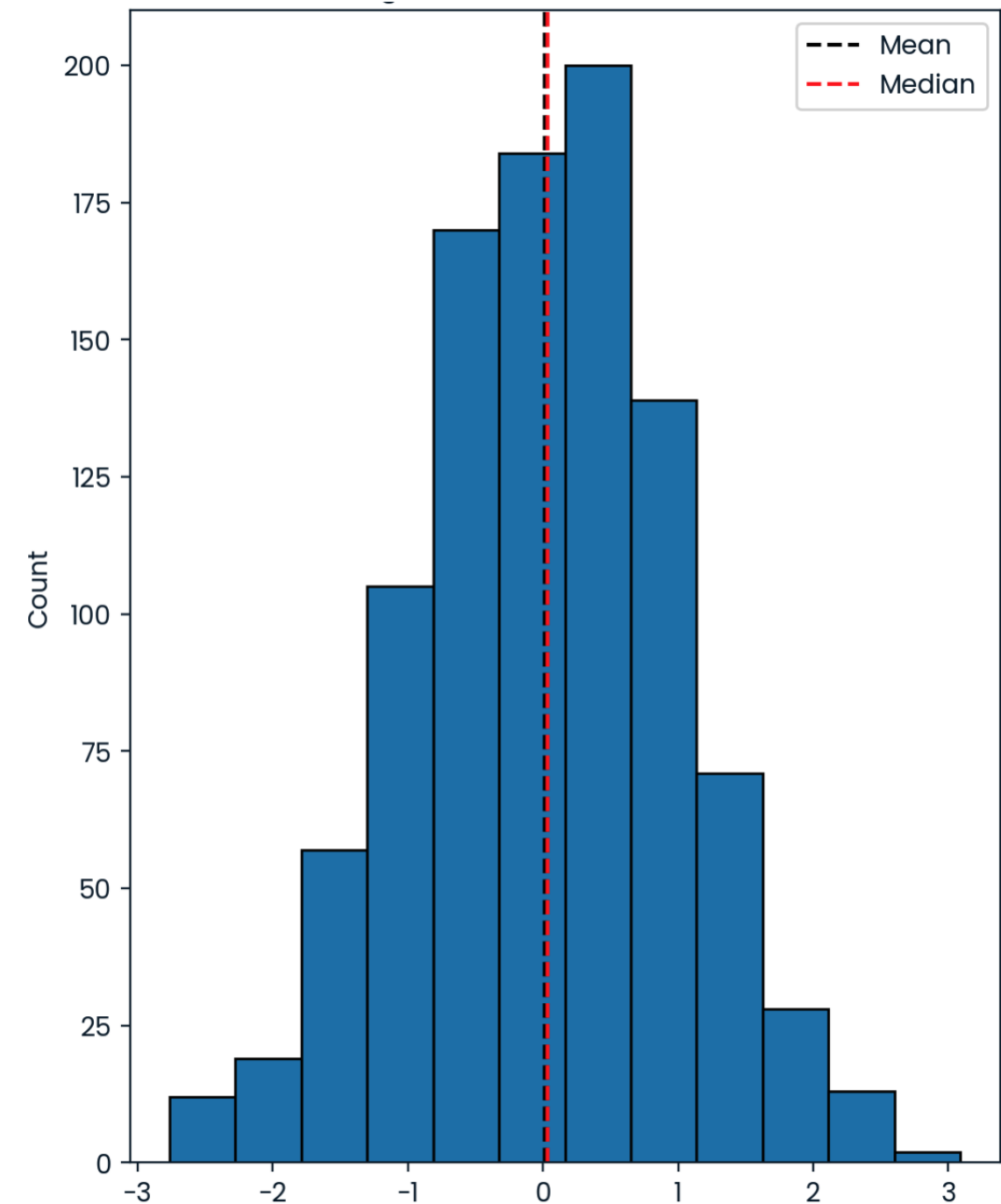
Mean: 16.5 → 13.2

Median: 18.9 → 18.1

Which measure to use?

```python
# Import matplotlib.pyplot with alias plt
import matplotlib.pyplot as plt

# Histogram of values
data['values'].hist()

# Show the plot
plt.show()
```
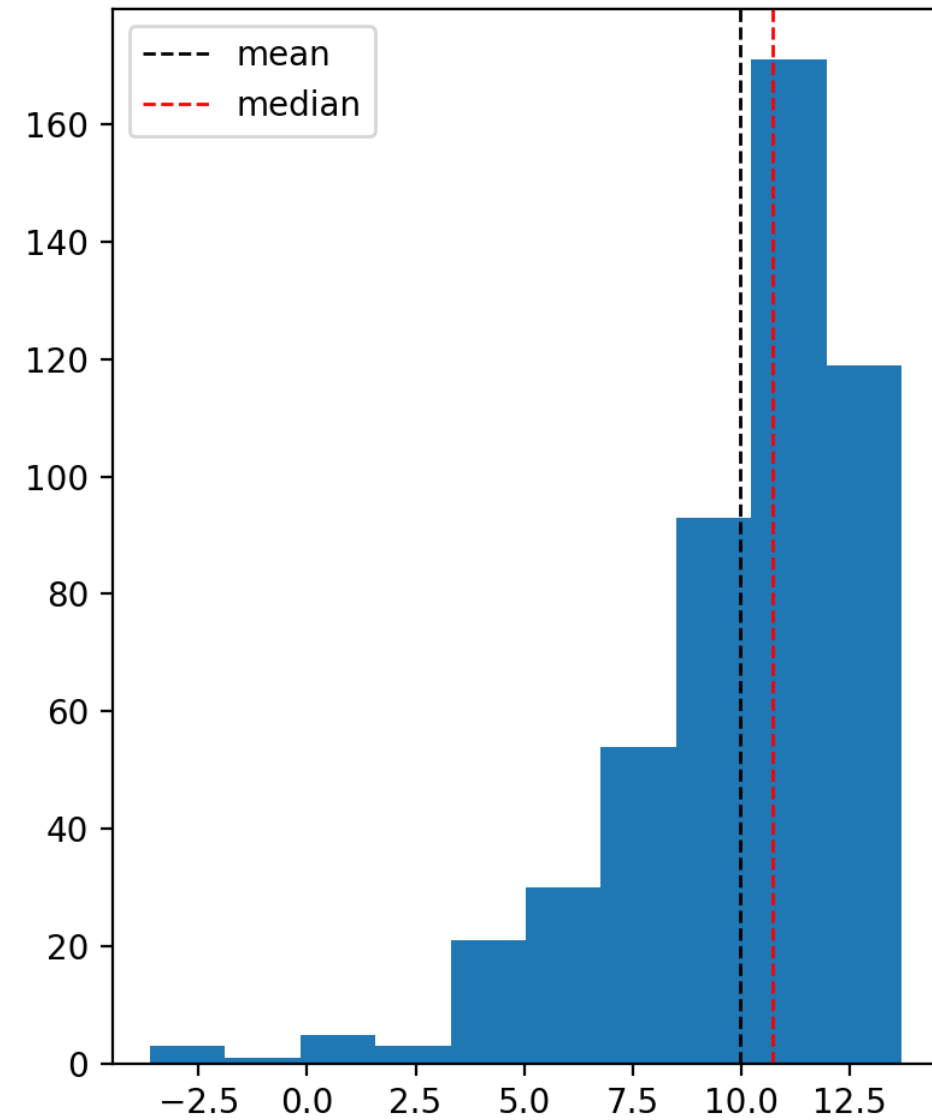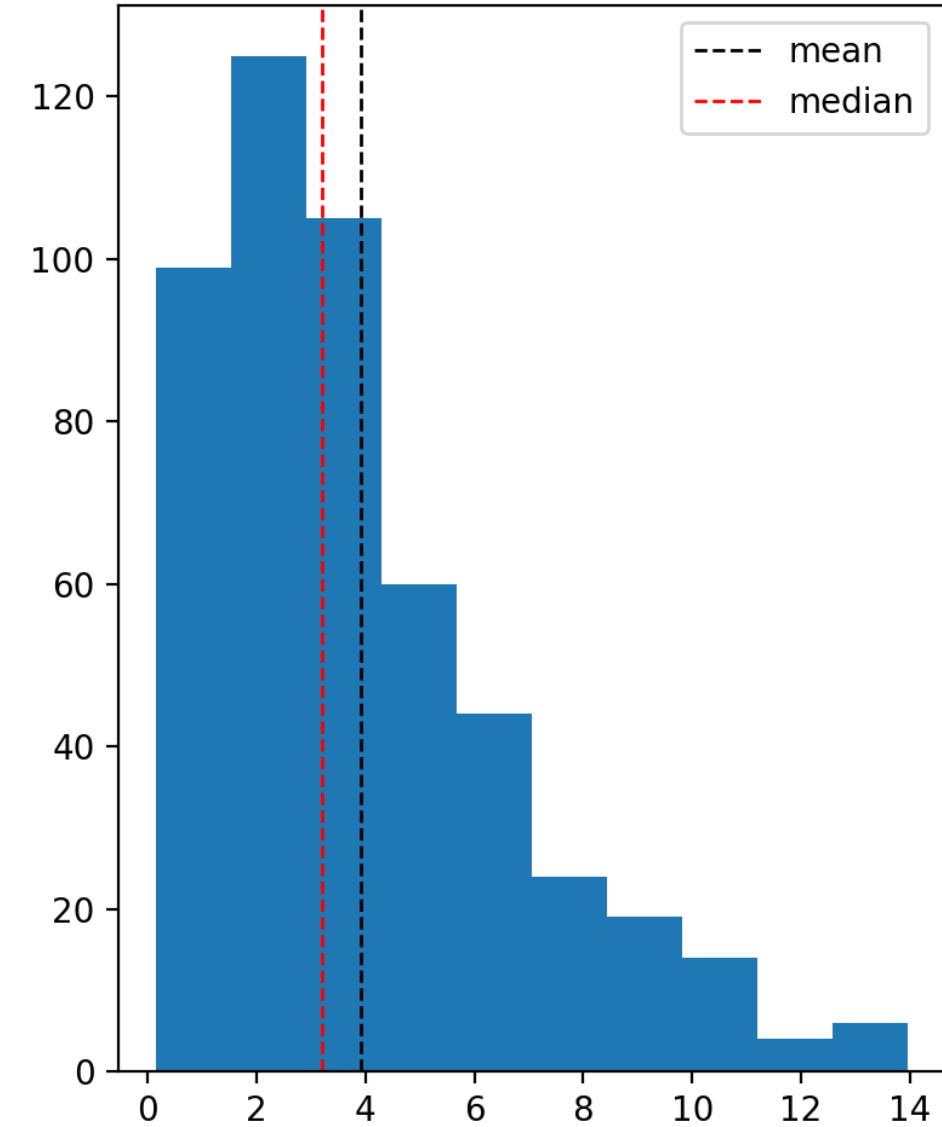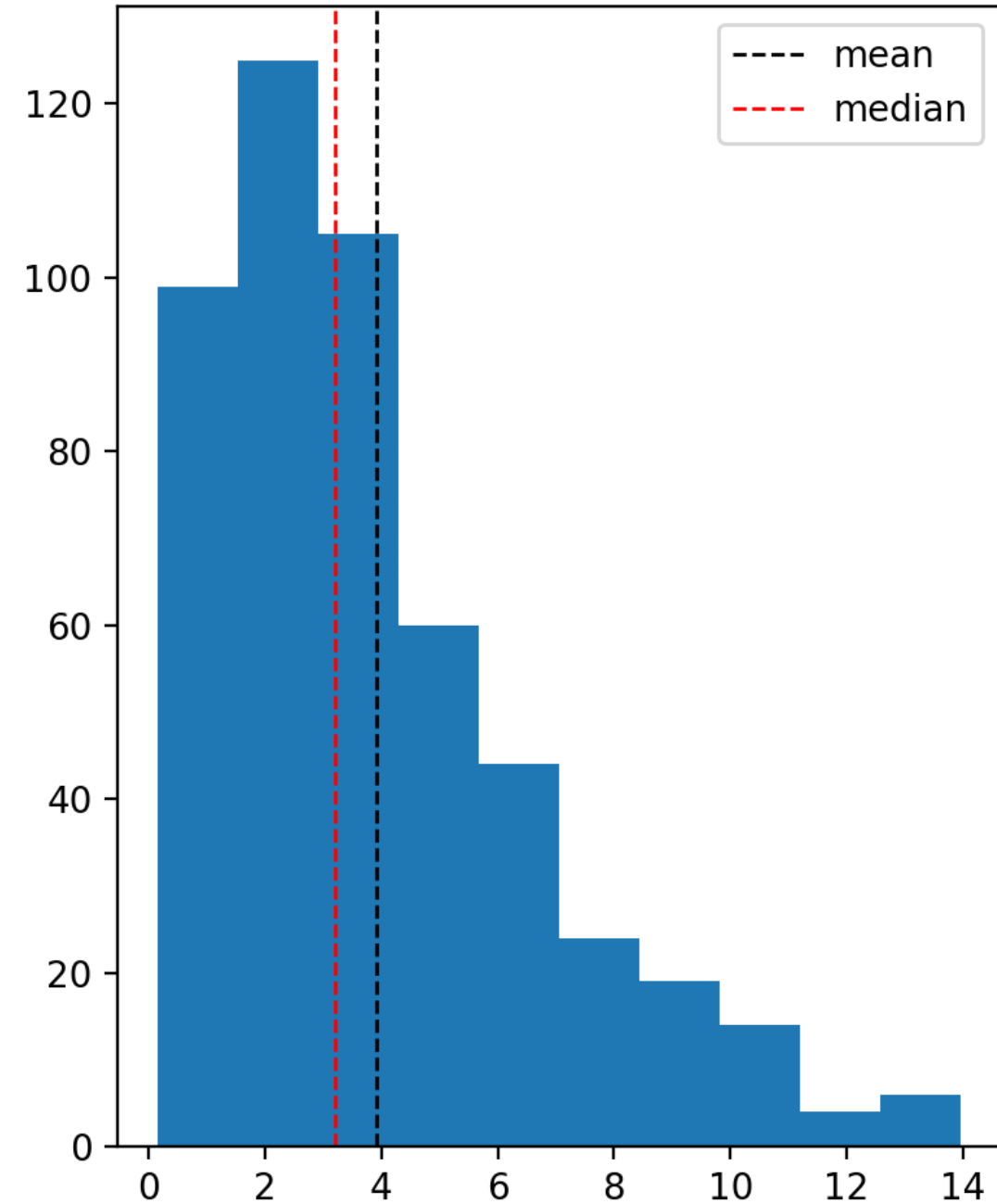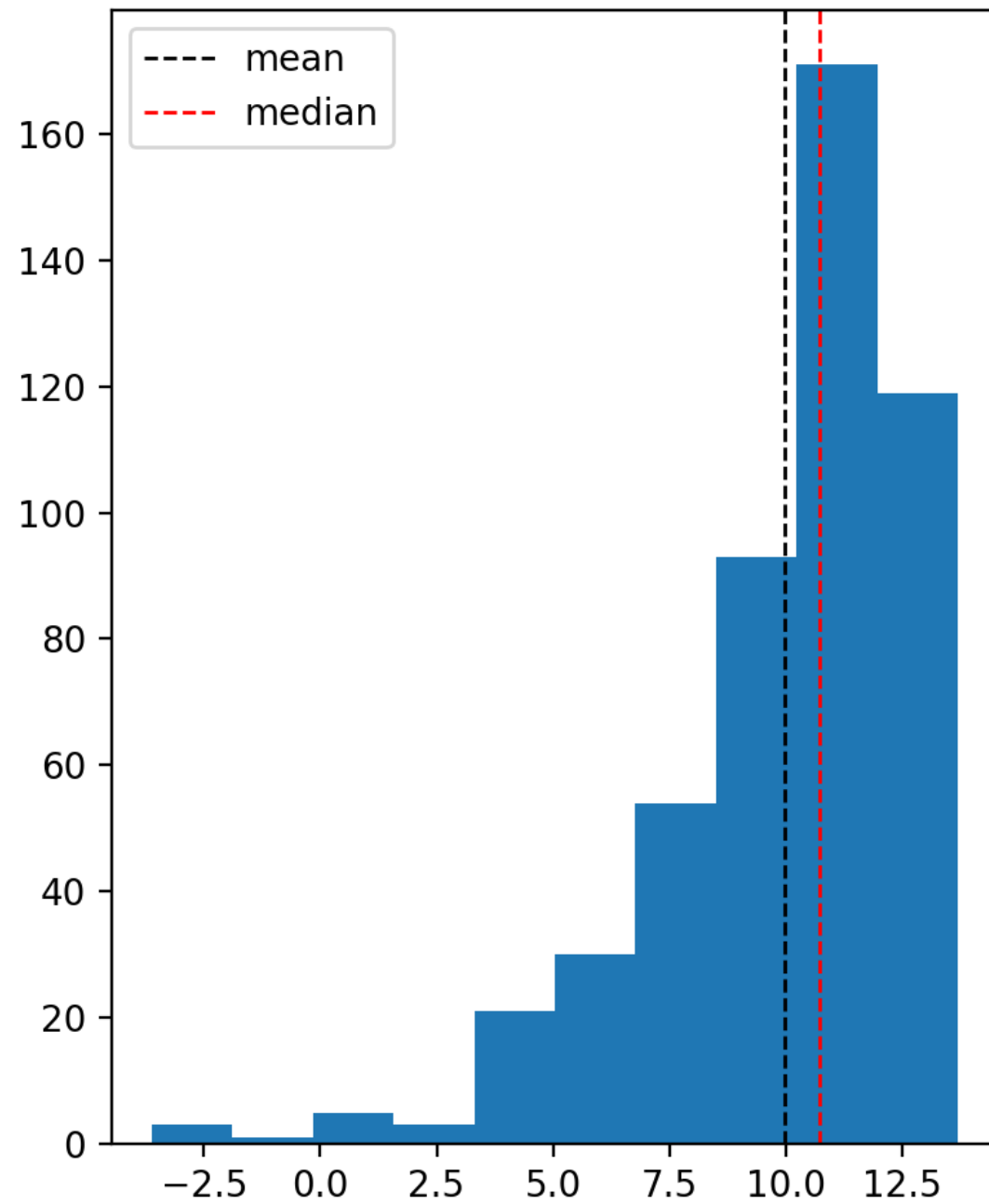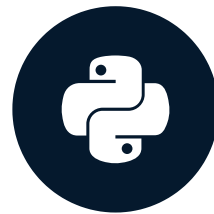
Skew

Which measure to use?

Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON

Measures of spread

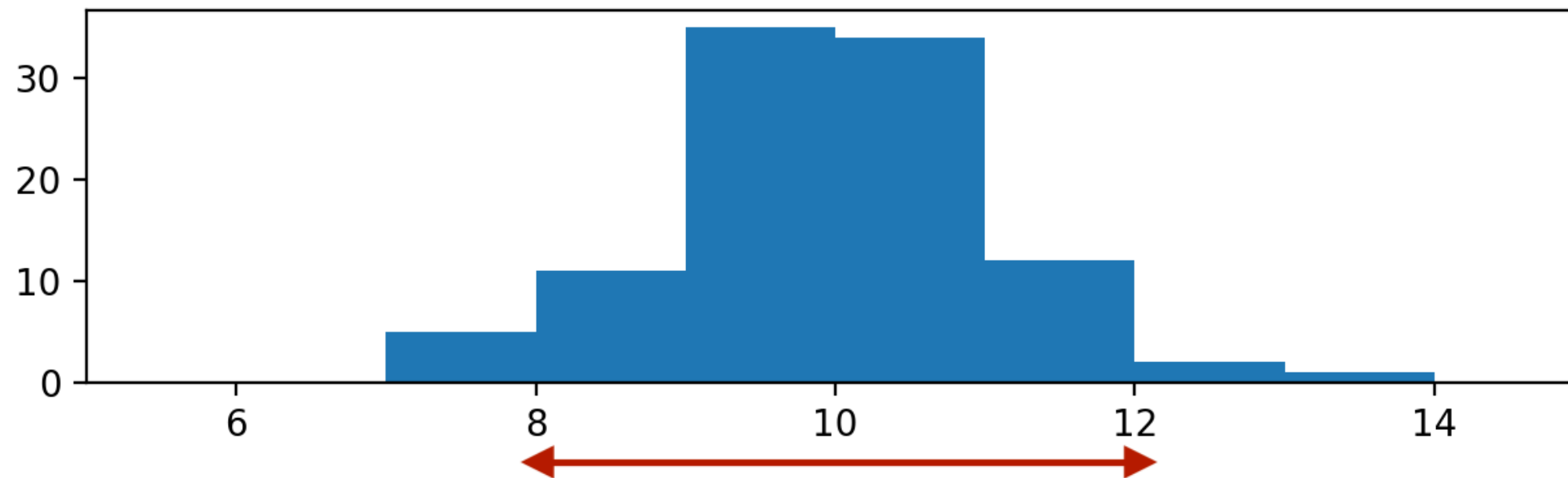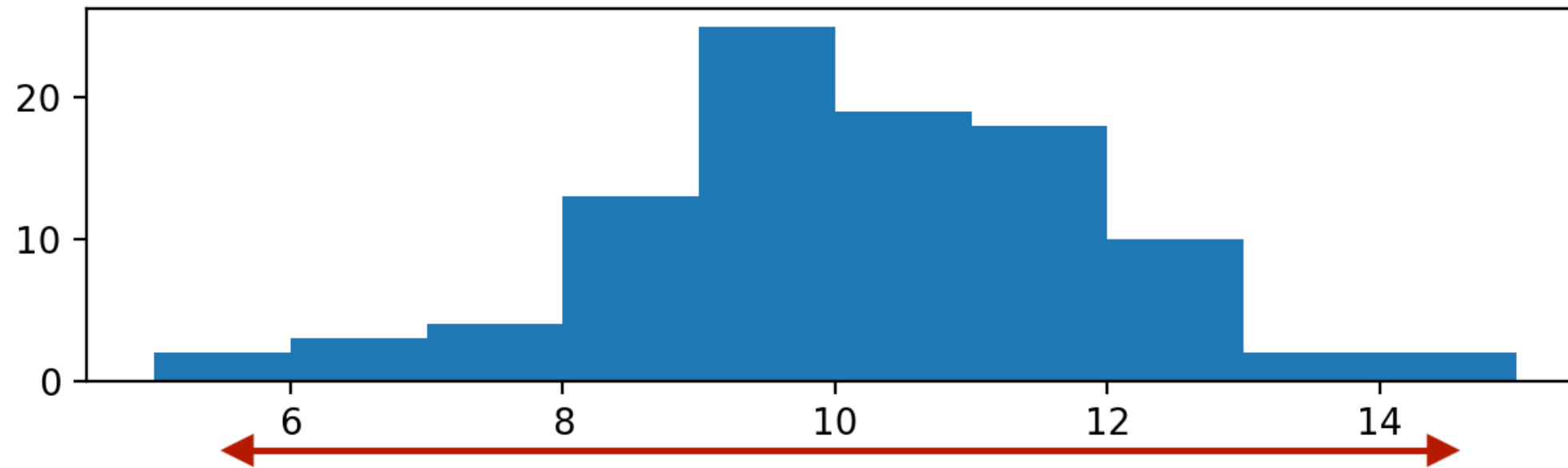## INTRODUCTION TO STATISTICS IN PYTHON
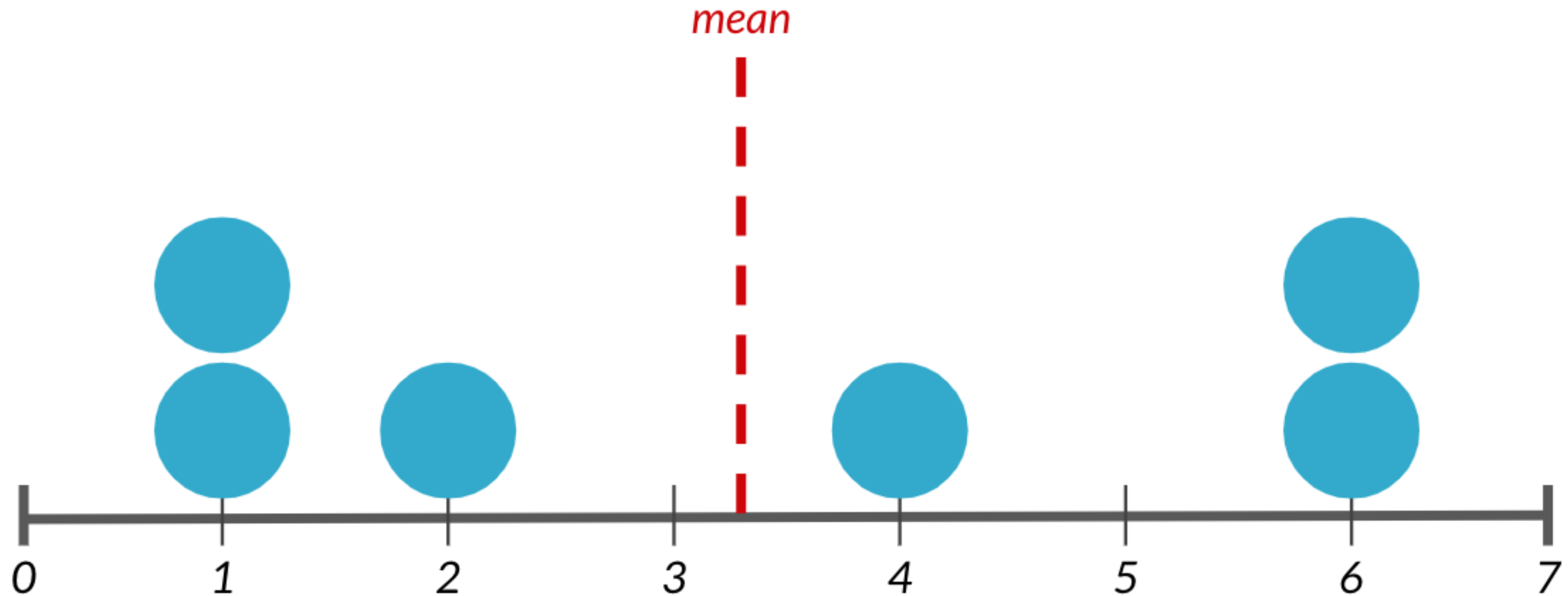
Maggie Matsui
Content Developer, DataCamp

What is spread?

Variance

Average distance from each data point to the data's mean

Variance

Average distance from each data point to the data's mean
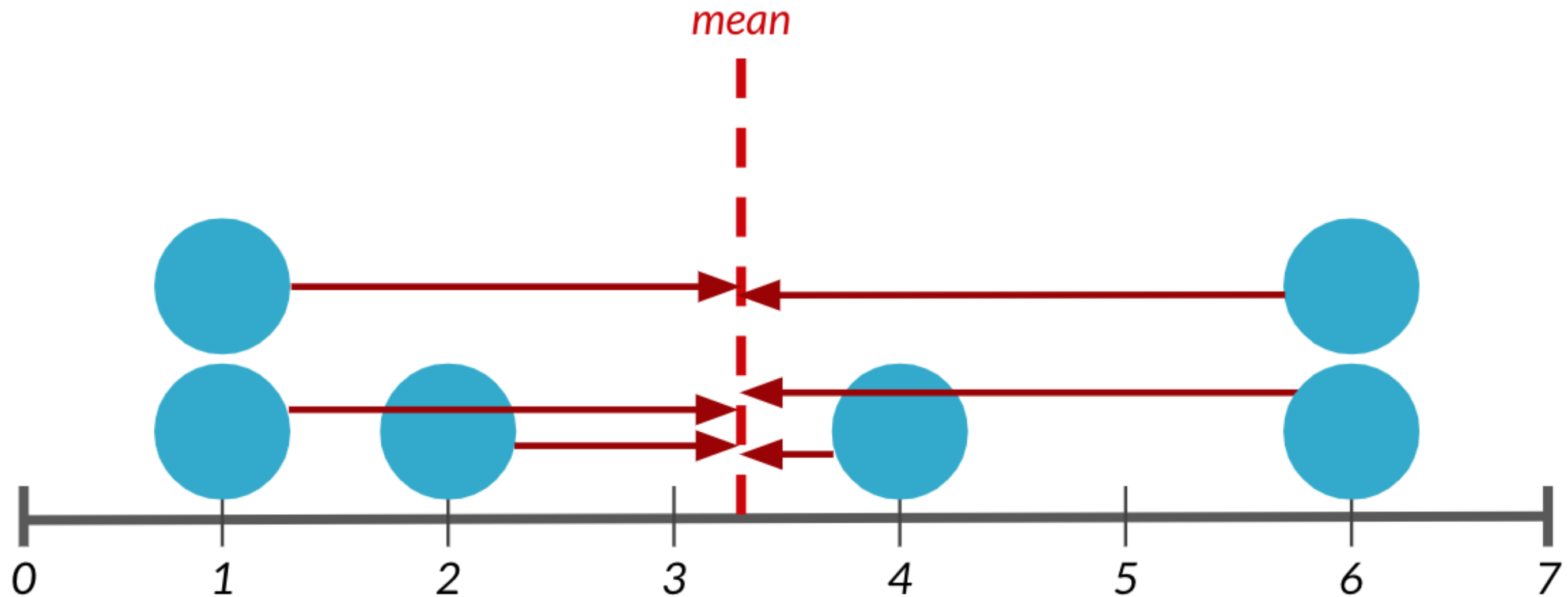
Calculating variance

1. Subtract mean from each data point

```python
dists = msleep['sleep_total'] -
        np.mean(msleep['sleep_total'])
print(dists)
```

```
0      1.666265
1      6.566265
2      3.966265
3      4.466265
4     -6.433735
      ...
```

2. Square each distance

```python
sq_dists = dists ** 2
print(sq_dists)
```

```
0      2.776439
1     43.115837
2     15.731259
3     19.947524
4     41.392945
      ...
```

Calculating variance

3. Sum squared distances

```python
sum_sq_dists = np.sum(sq_dists)
print(sum_sq_dists)
```

```
1624.065542
```

4. Divide by number of data points - 1

```python
variance = sum_sq_dists / (83 - 1)
print(variance)
```

```
19.805677
```

Use `np.var()`

```python
np.var(msleep['sleep_total'], ddof=1)
```

```
19.805677
```

*Without `ddof=1`, **population variance** is calculated instead of **sample variance**:*

```python
np.var(msleep['sleep_total'])
```

```
19.567055
```

Standard deviation

```python
np.sqrt(np.var(msleep['sleep_total'], ddof=1))
```

```
4.450357
```

```python
np.std(msleep['sleep_total'], ddof=1)
```

```
4.450357
```

Mean absolute deviation

```python
dists = msleep['sleep_total'] - np.mean(msleep['sleep_total'])
np.mean(np.abs(dists))
```

```
3.566701
```

Standard deviation vs. mean absolute deviation

- Standard deviation squares distances, penalizing longer distances more than shorter ones.

- Mean absolute deviation penalizes each distance equally.

- One isn't better than the other, but SD is more common than MAD.

Quantiles

```python
np.quantile(msleep['sleep_total'], 0.5)
```
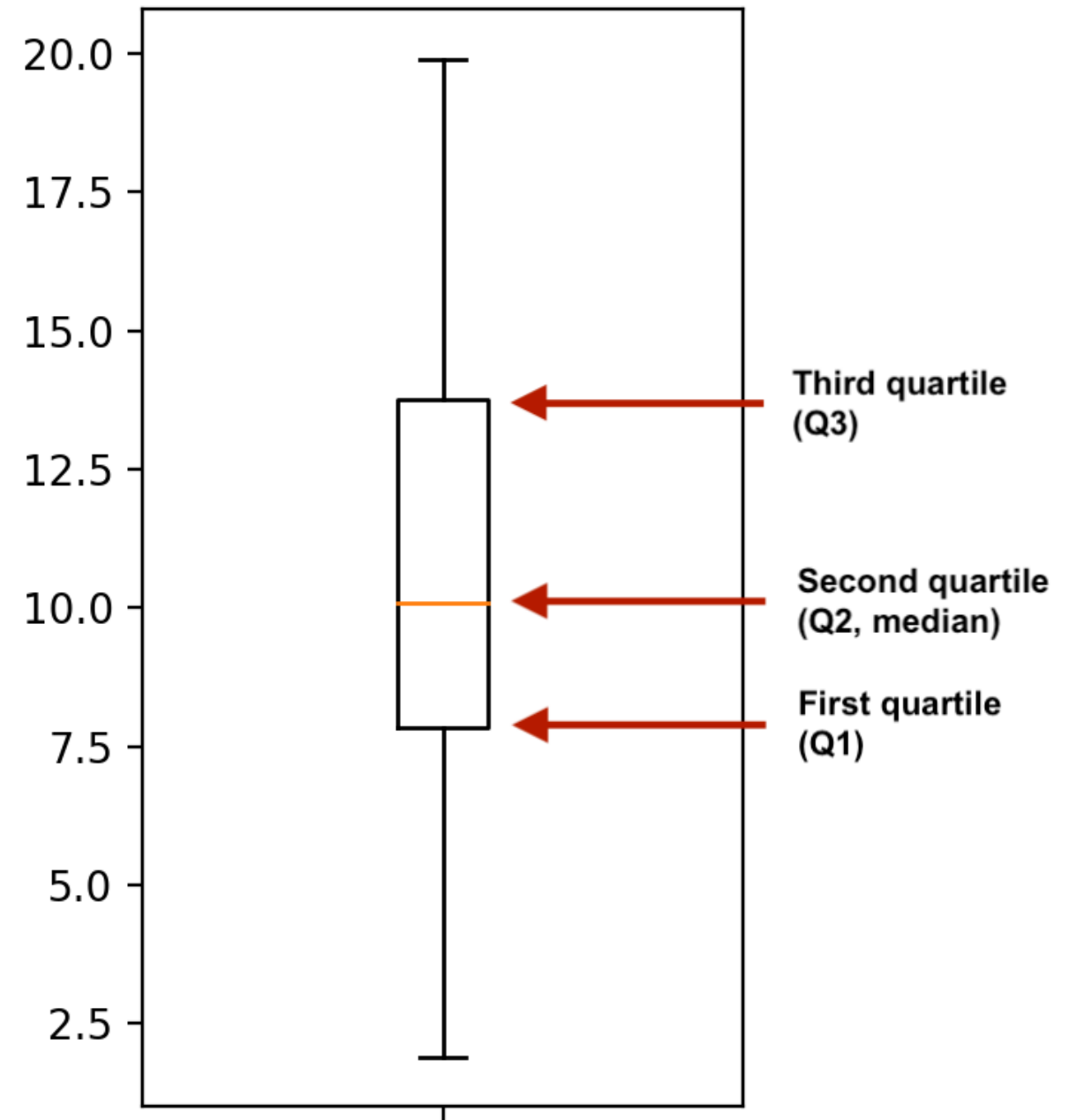
```
10.1
```

0.5 quantile = median

Quartiles:

```python
np.quantile(msleep['sleep_total'], [0, 0.25, 0.5, 0.75, 1])
```

```
array([ 1.9 ,  7.85, 10.1 , 13.75, 19.9 ])
```

Boxplots use quartiles

```python
import matplotlib.pyplot as plt
plt.boxplot(msleep['sleep_total'])
plt.show()
```

Quantiles using np.linspace()

```python
np.quantile(msleep['sleep_total'], [0, 0.2, 0.4, 0.6, 0.8, 1])
```

```
array([ 1.9 ,  6.24,  9.48, 11.14, 14.4 , 19.9 ])
```

```python
np.linspace(start, stop, num)
```

```python
np.quantile(msleep['sleep_total'], np.linspace(0, 1, 5))
```

```
array([ 1.9 ,  7.85, 10.1 , 13.75, 19.9 ])
```

Interquartile range (IQR)

Height of the box in a boxplot

```python
np.quantile(msleep['sleep_total'], 0.75) - np.quantile(msleep['sleep_total'], 0.25)
```

```
5.9
```

```python
from scipy.stats import iqr
iqr(msleep['sleep_total'])
```

```
5.9
```

Outliers

Outlier: data point that is substantially different from the others

How do we know what a substantial difference is? A data point is an outlier if:

- $\text{data} < \text{Q1} - 1.5 \times \text{IQR}$ or
- $\text{data} > \text{Q3} + 1.5 \times \text{IQR}$

Finding outliers

```python
from scipy.stats import iqr
iqr = iqr(msleep['bodywt'])
lower_threshold = np.quantile(msleep['bodywt'], 0.25) - 1.5 * iqr
upper_threshold = np.quantile(msleep['bodywt'], 0.75) + 1.5 * iqr
```

```python
msleep[(msleep['bodywt'] < lower_threshold) | (msleep['bodywt'] > upper_threshold)]
```

```
              name    vore  sleep_total    bodywt
4              Cow   herbi          4.0   600.000
20   Asian elephant  herbi          3.9  2547.000
22            Horse  herbi          2.9   521.000
...
```

All in one go

```
msleep['bodywt'].describe()
```

```
count      83.000000
mean      166.136349
std       786.839732
min         0.005000
25%         0.174000
50%         1.670000
75%        41.750000
max      6654.000000
Name: bodywt, dtype: float64
```

Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON