

# The normal distribution

INTRODUCTION TO STATISTICS IN PYTHON

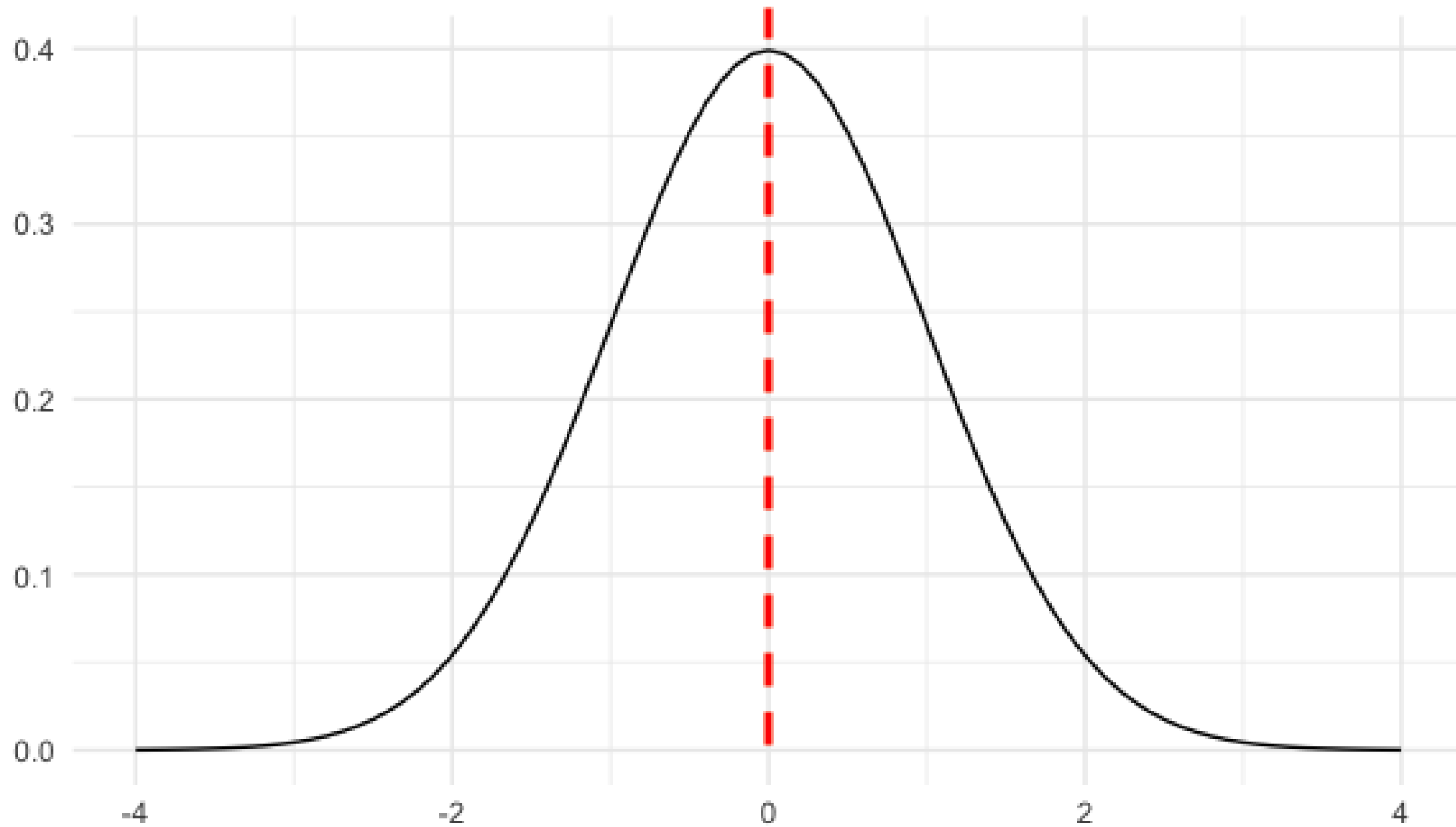


**Maggie Matsui**  
Content Developer, DataCamp

# What is the normal distribution?



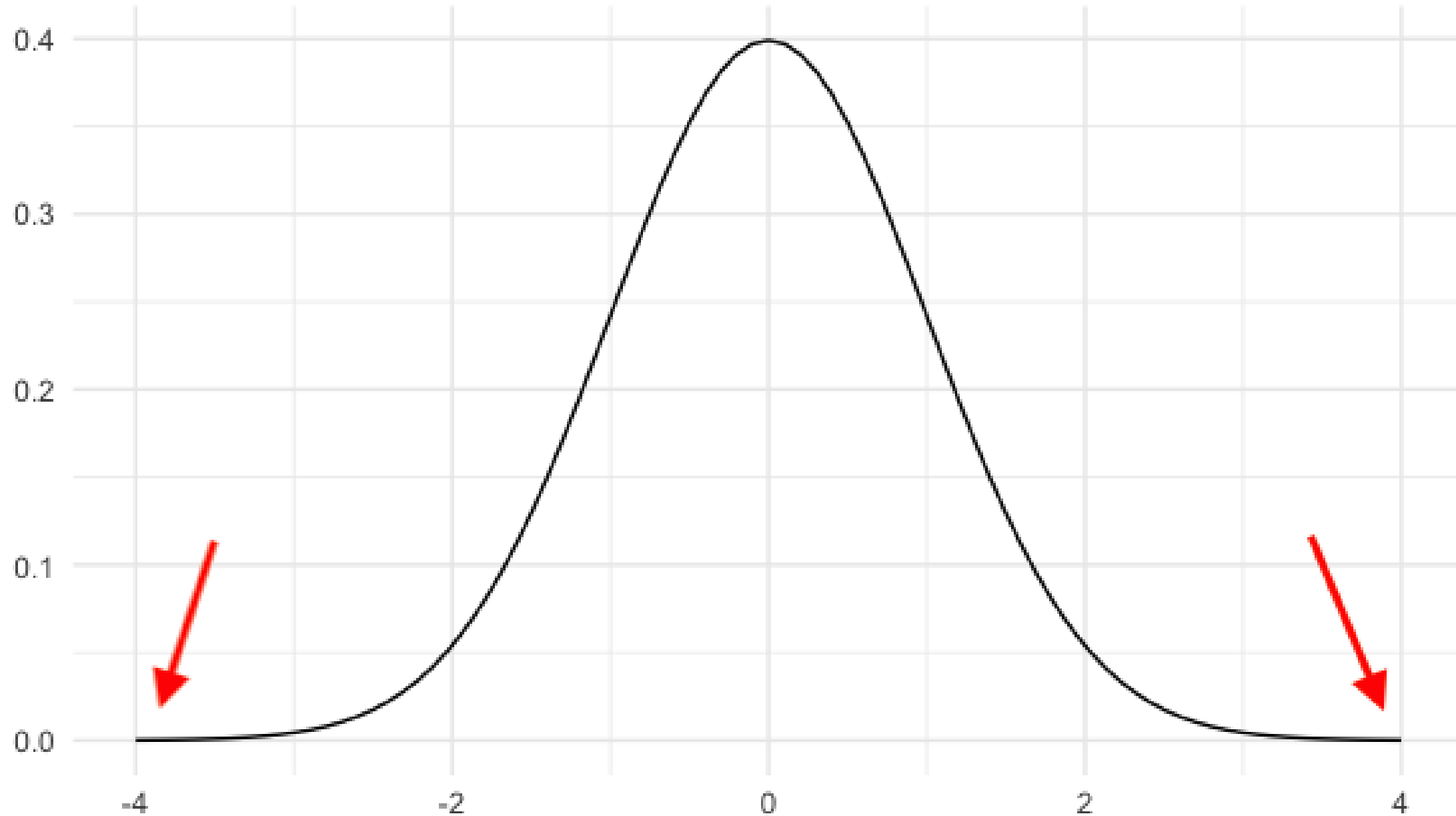
# Symmetrical



# Area = 1



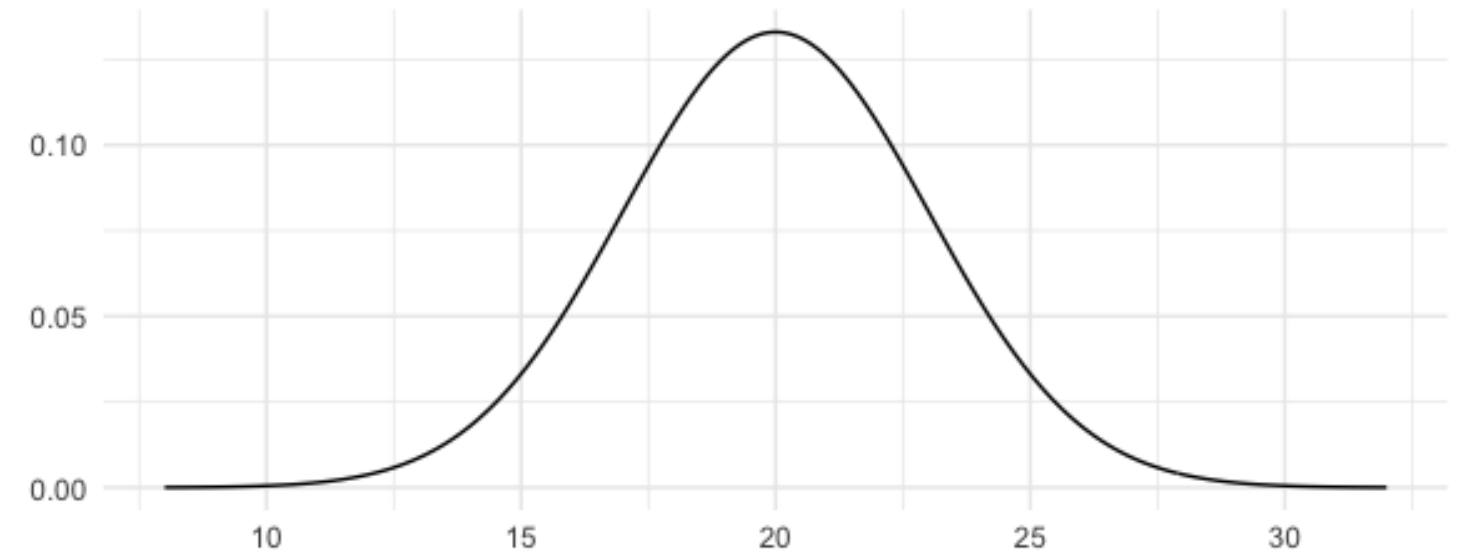
# Curve never hits 0



# Described by mean and standard deviation

Mean: 20

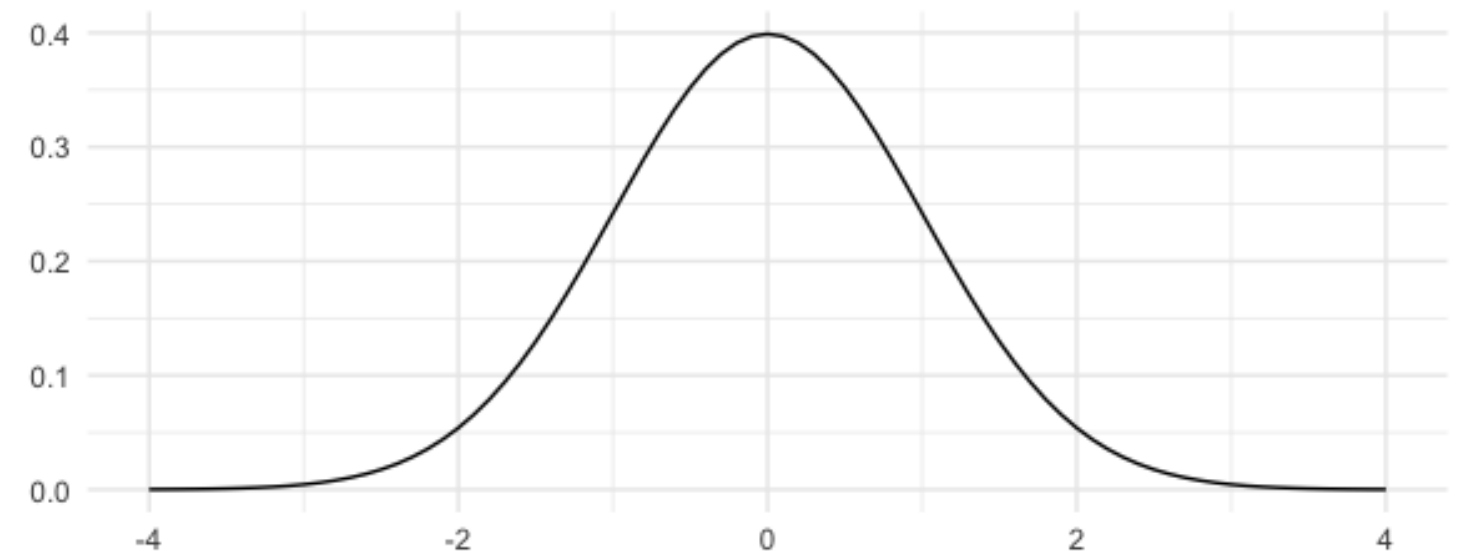
Standard deviation: 3



*Standard normal distribution*

Mean: 0

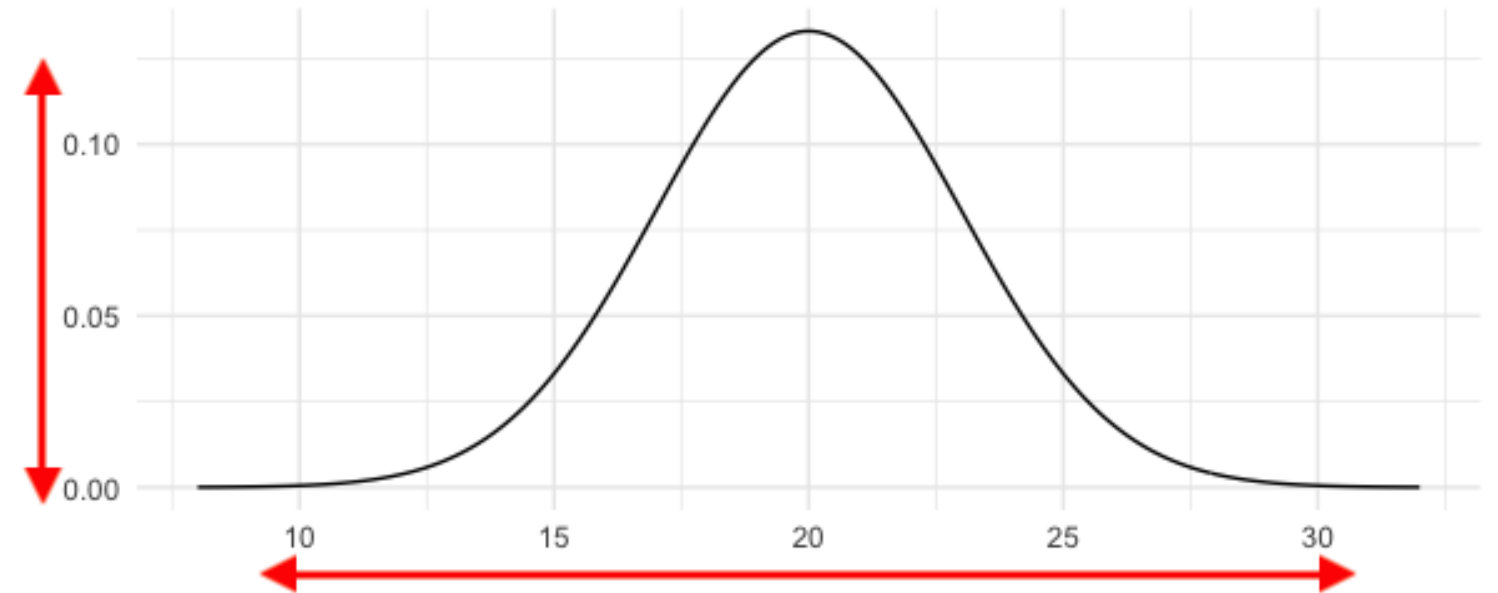
Standard deviation: 1



# Described by mean and standard deviation

Mean: 20

Standard deviation: 3



*Standard normal distribution*

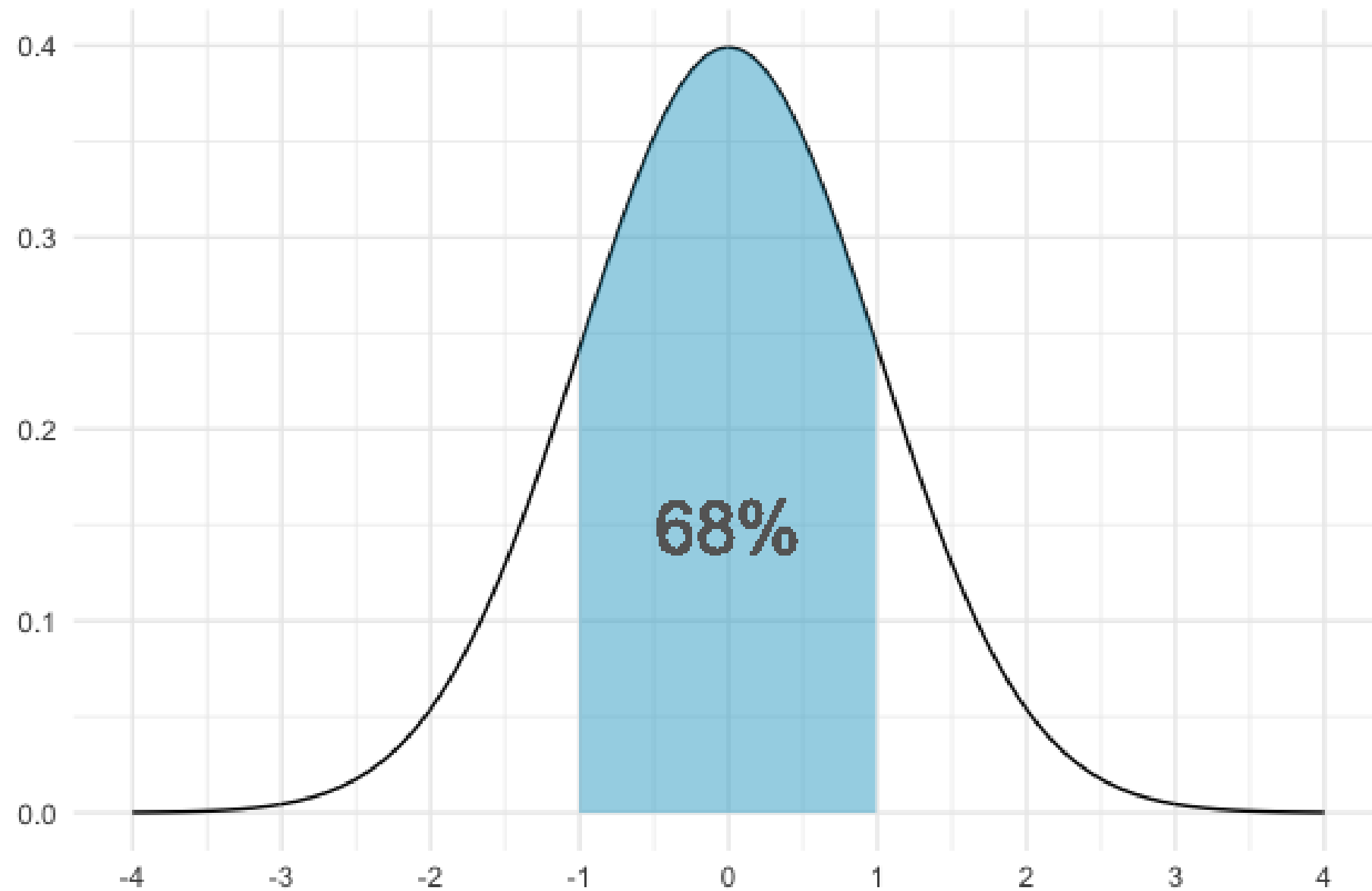
Mean: 0

Standard deviation: 1



# Areas under the normal distribution

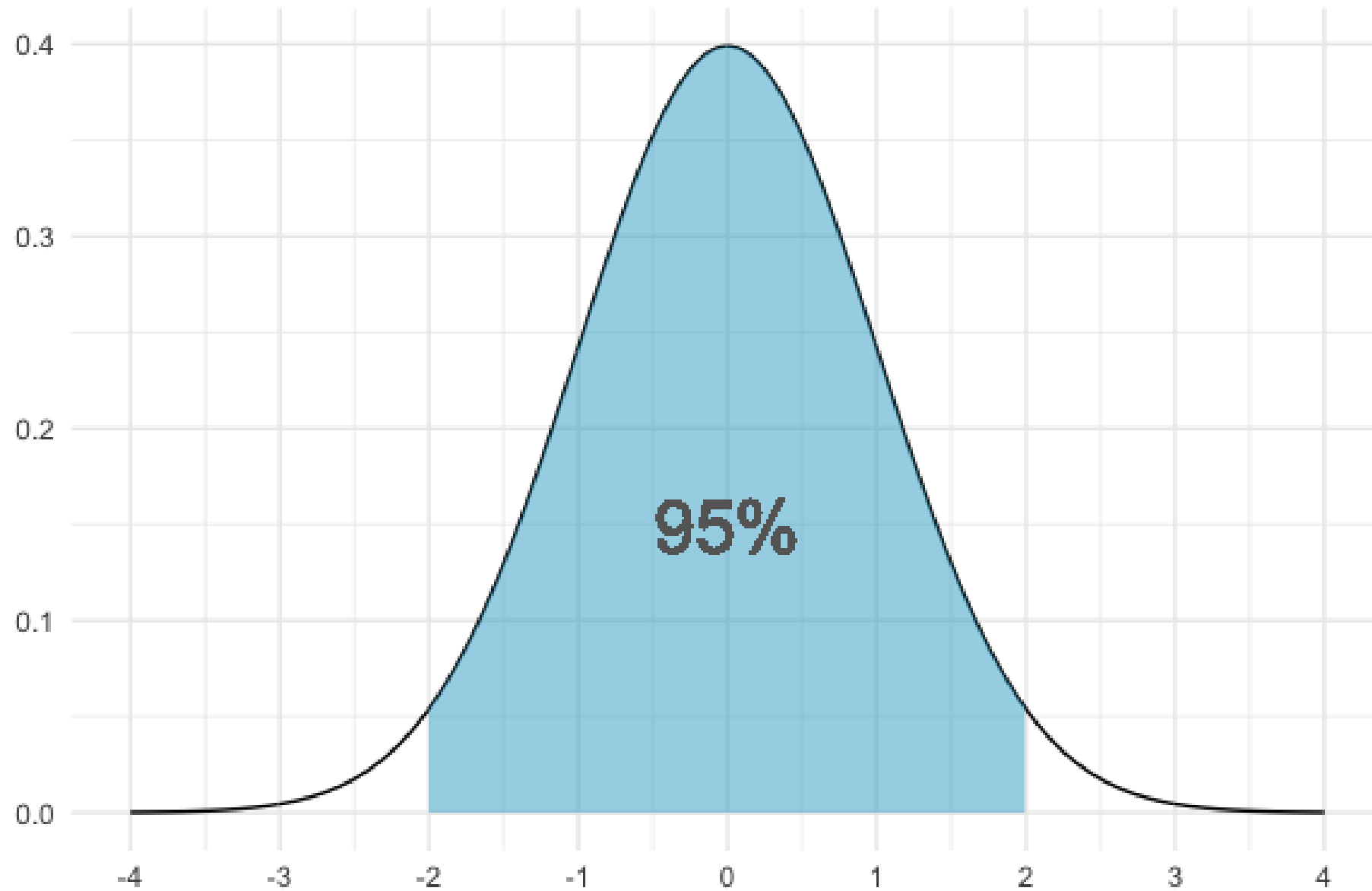
68% falls within 1 standard deviation





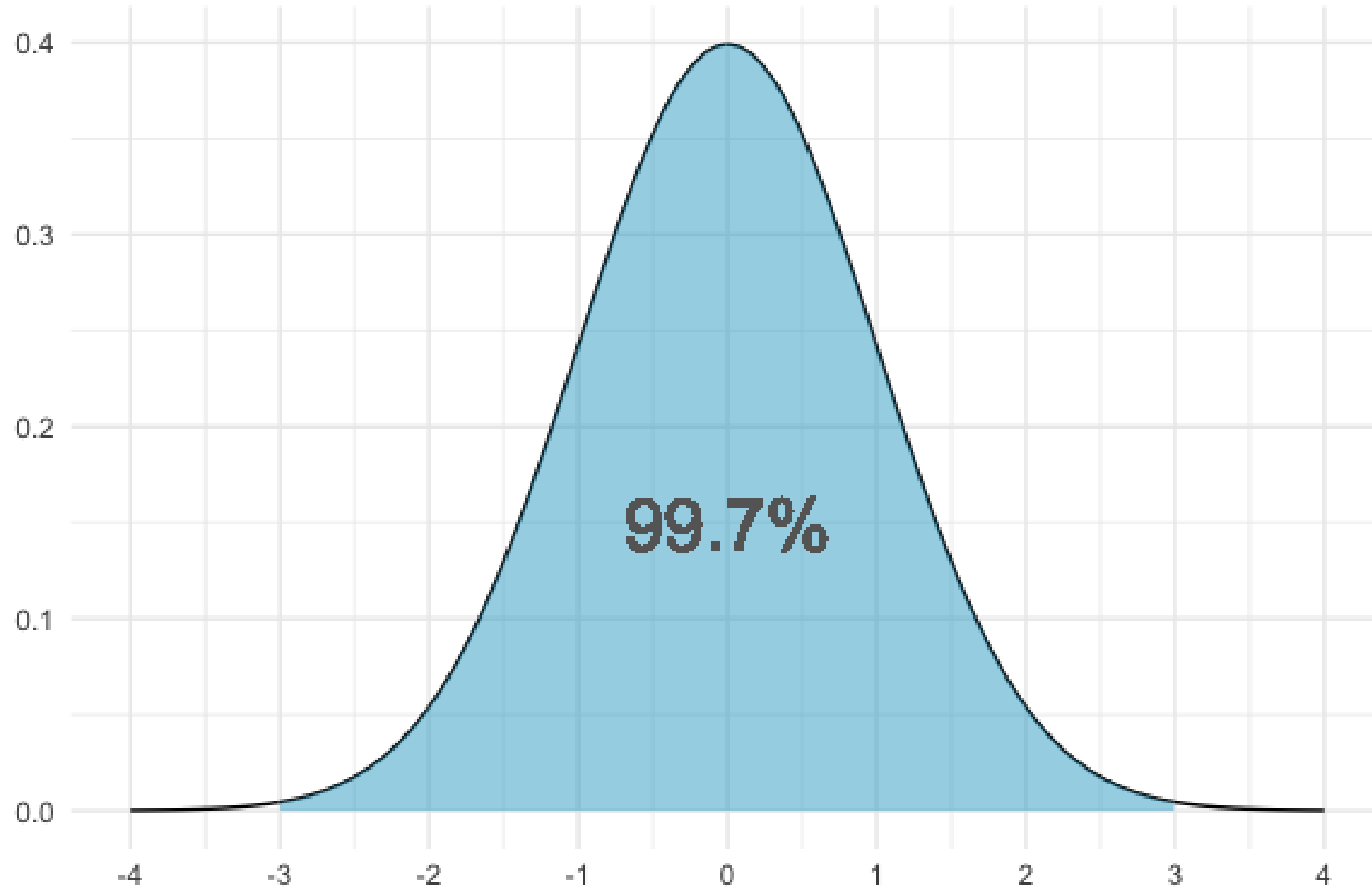
# Areas under the normal distribution

95% falls within 2 standard deviations



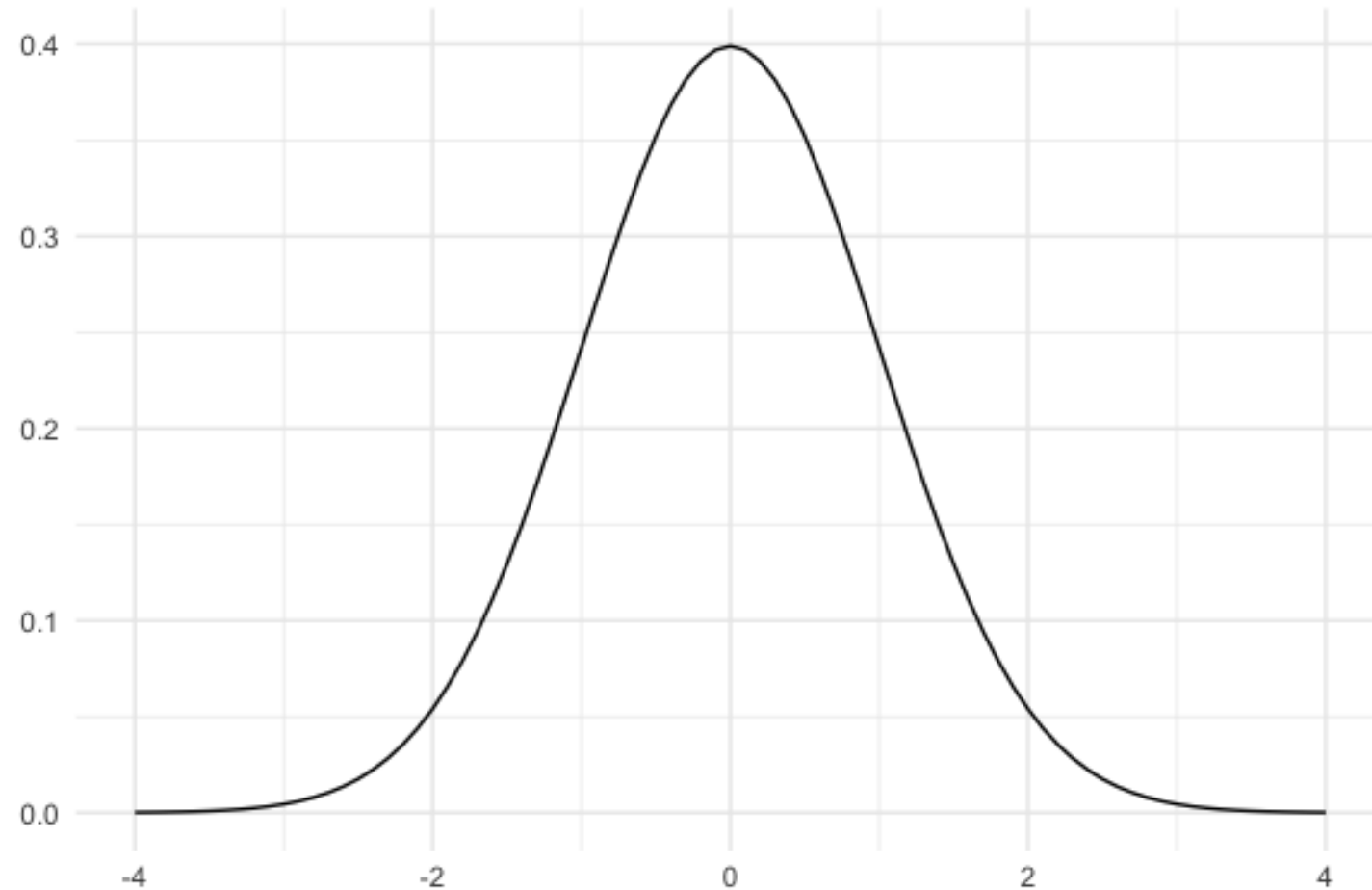
# Areas under the normal distribution

99.7% falls within 3 standard deviations

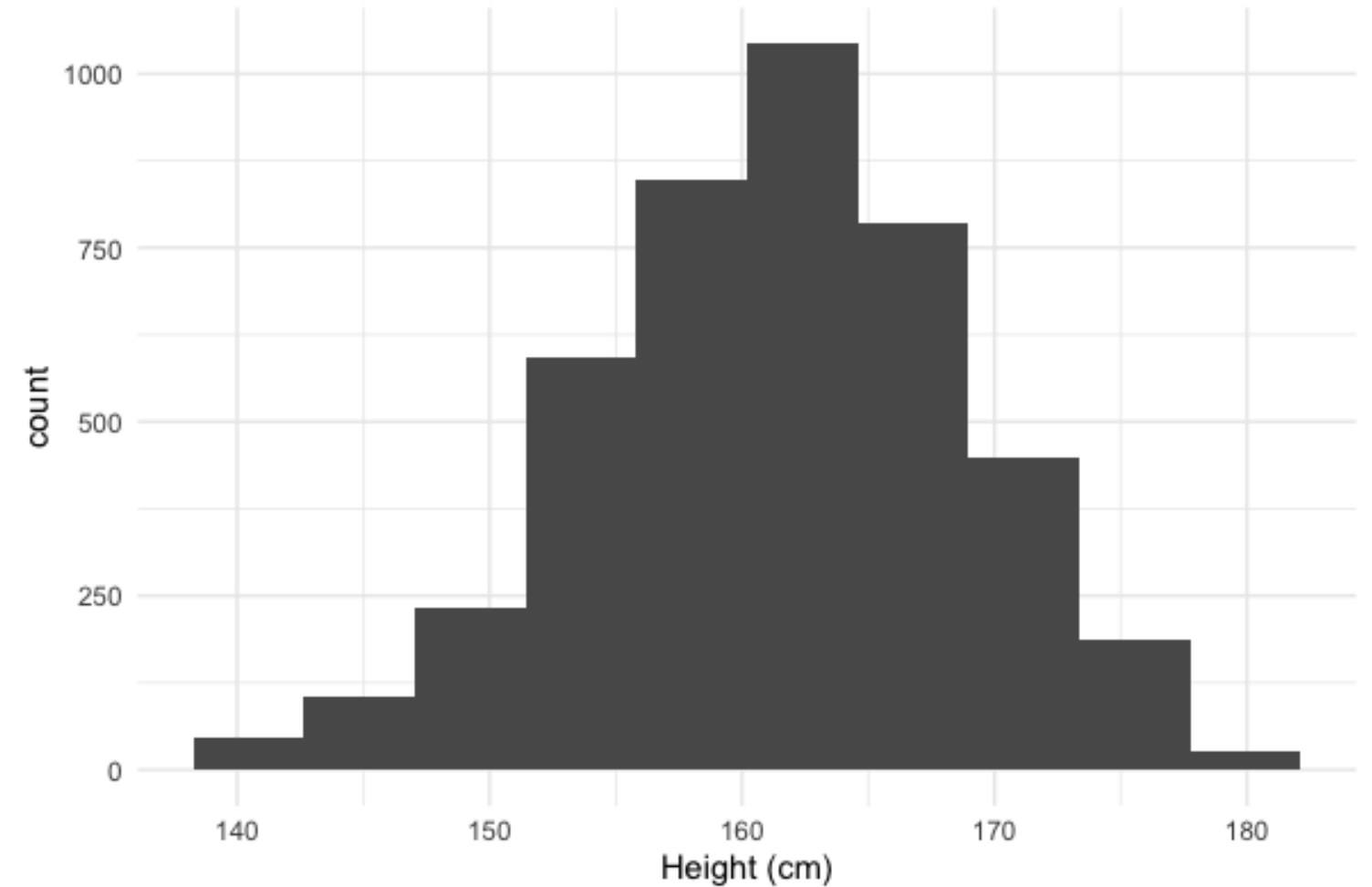


# Lots of histograms look normal

## Normal distribution



## Women's heights from NHANES



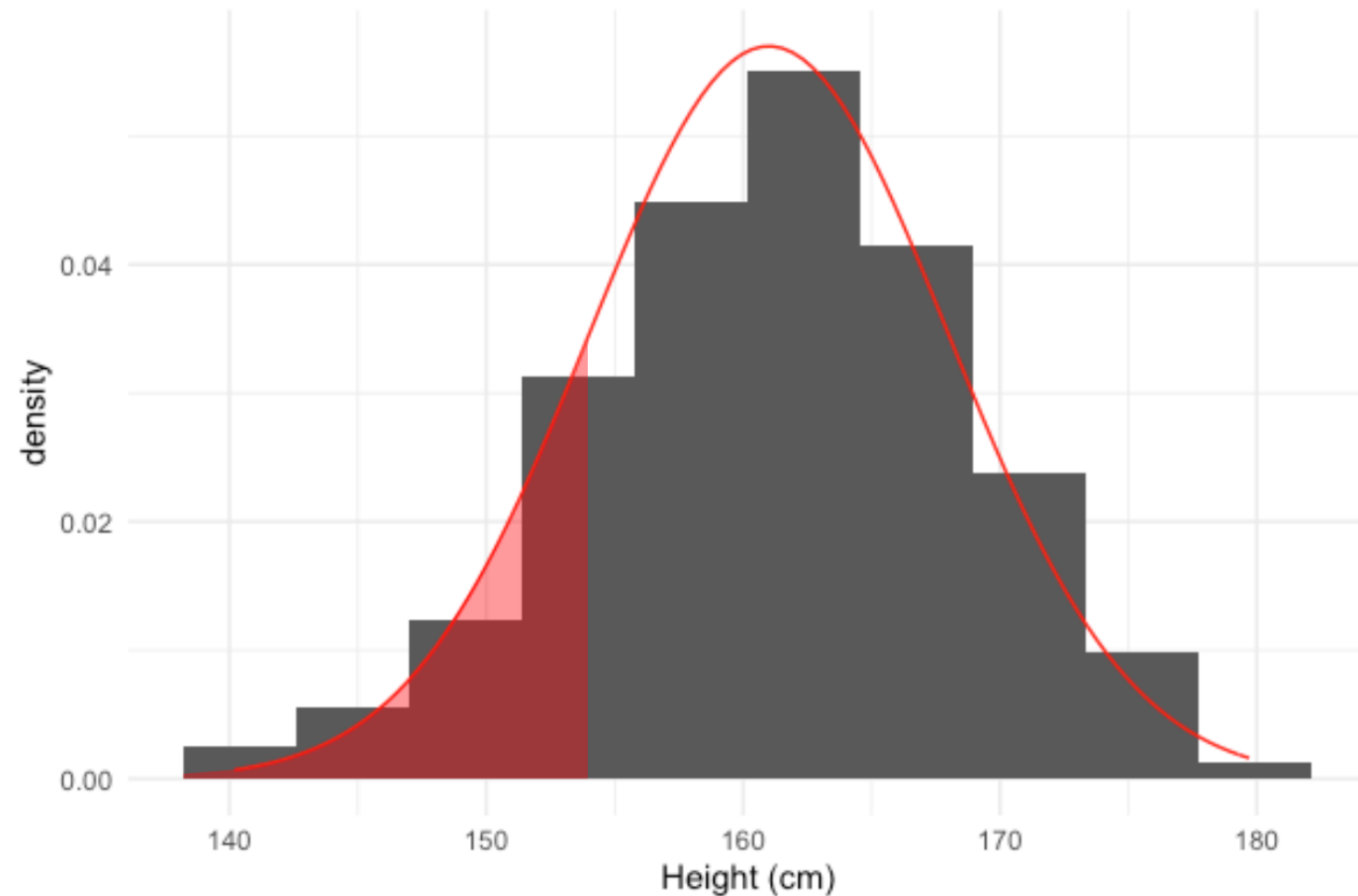
***Mean: 161 cm***

***Standard deviation: 7 cm***

# Approximating data with the normal distribution



# What percent of women are shorter than 154 cm?

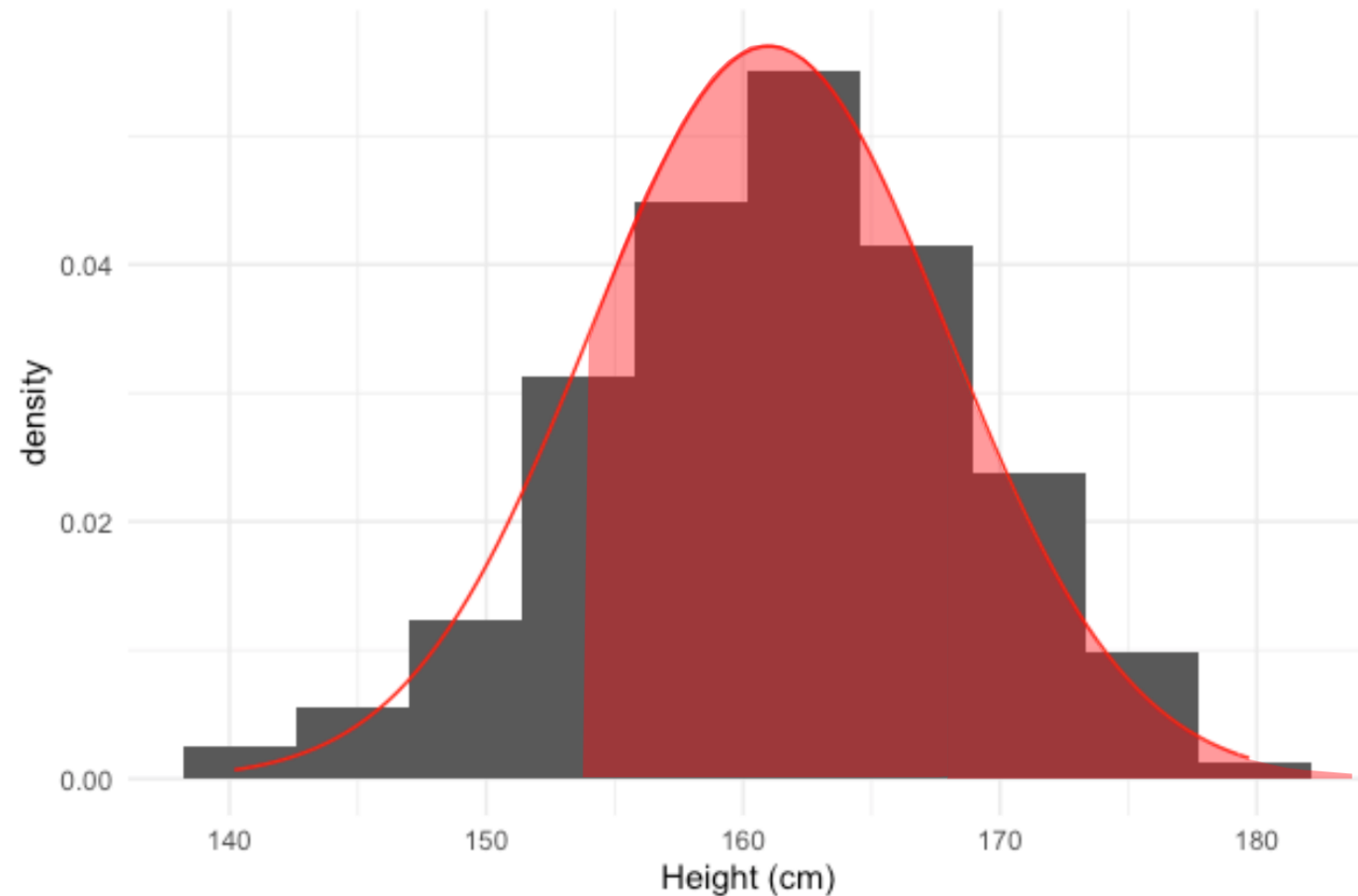


```
from scipy.stats import norm  
norm.cdf(154, 161, 7)
```

```
0.158655
```

*16% of women in the survey are shorter than 154 cm*

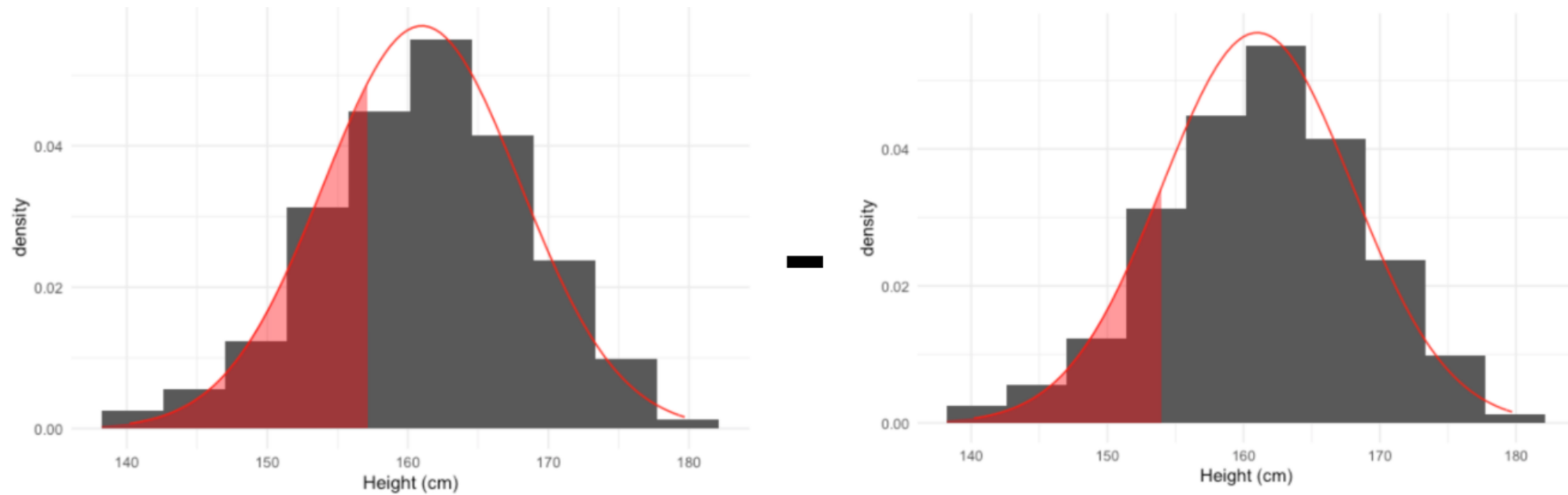
# What percent of women are taller than 154 cm?



```
from scipy.stats import norm  
1 - norm.cdf(154, 161, 7)
```

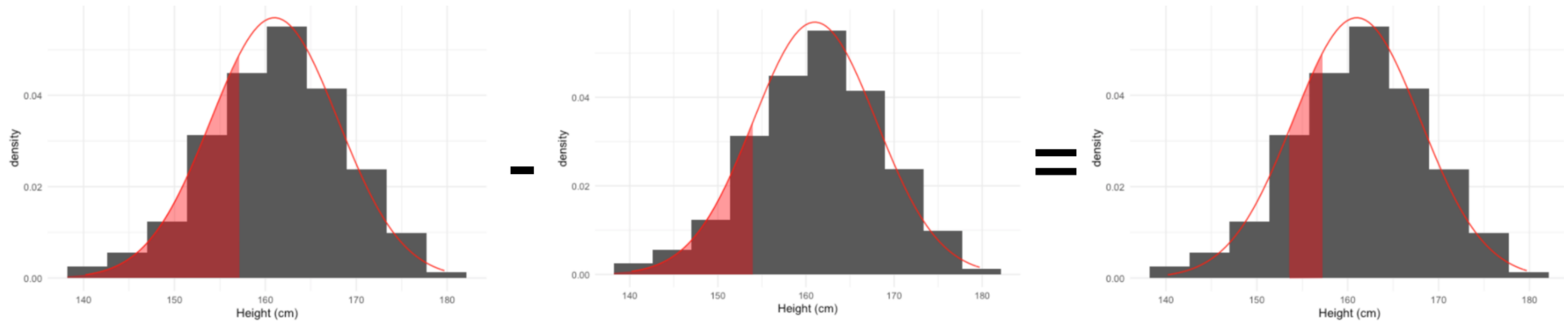
0.841345

# What percent of women are 154-157 cm?



```
norm.cdf(157, 161, 7) - norm.cdf(154, 161, 7)
```

# What percent of women are 154-157 cm?

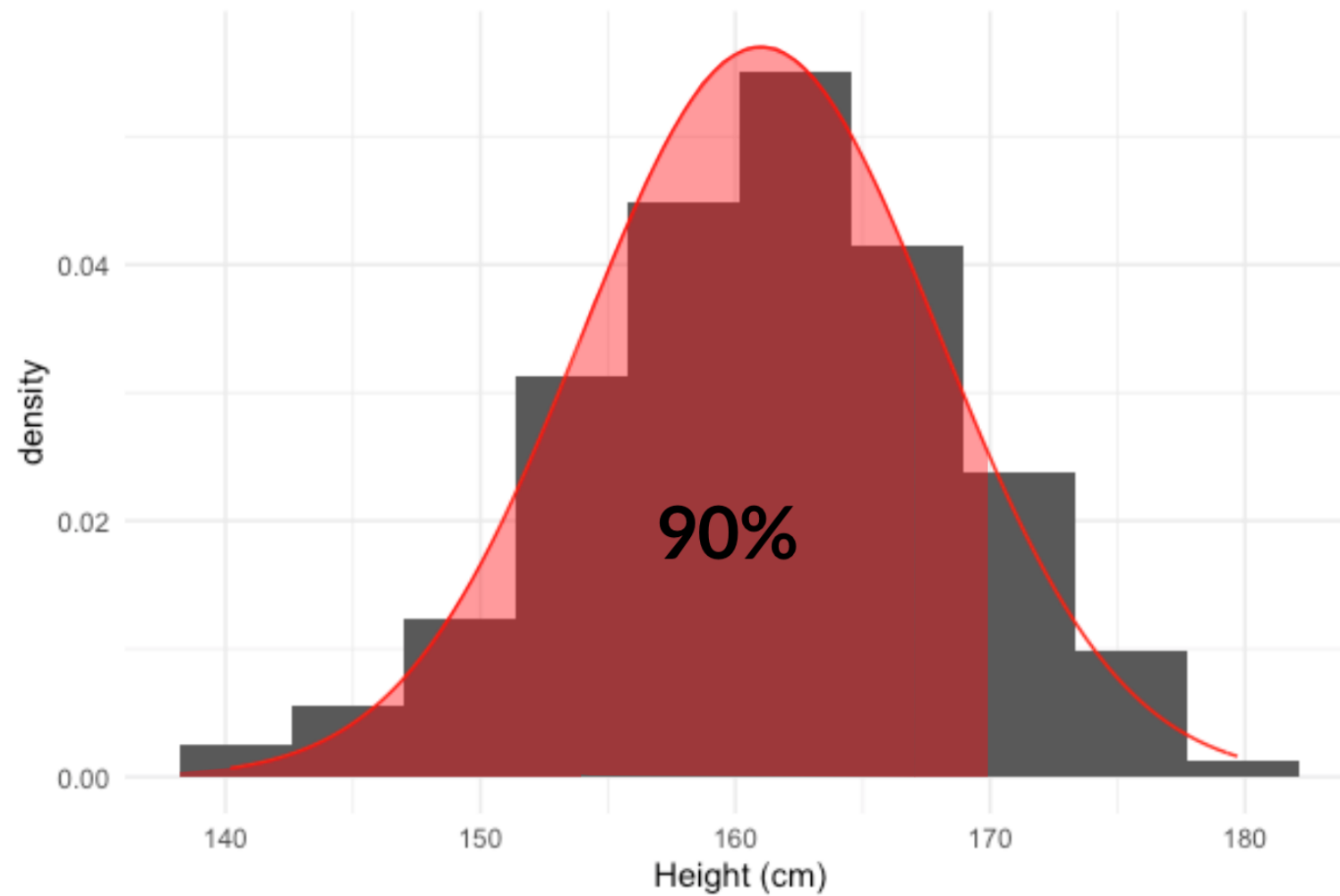


```
norm.cdf(157, 161, 7) - norm.cdf(154, 161, 7)
```

```
0.1252
```



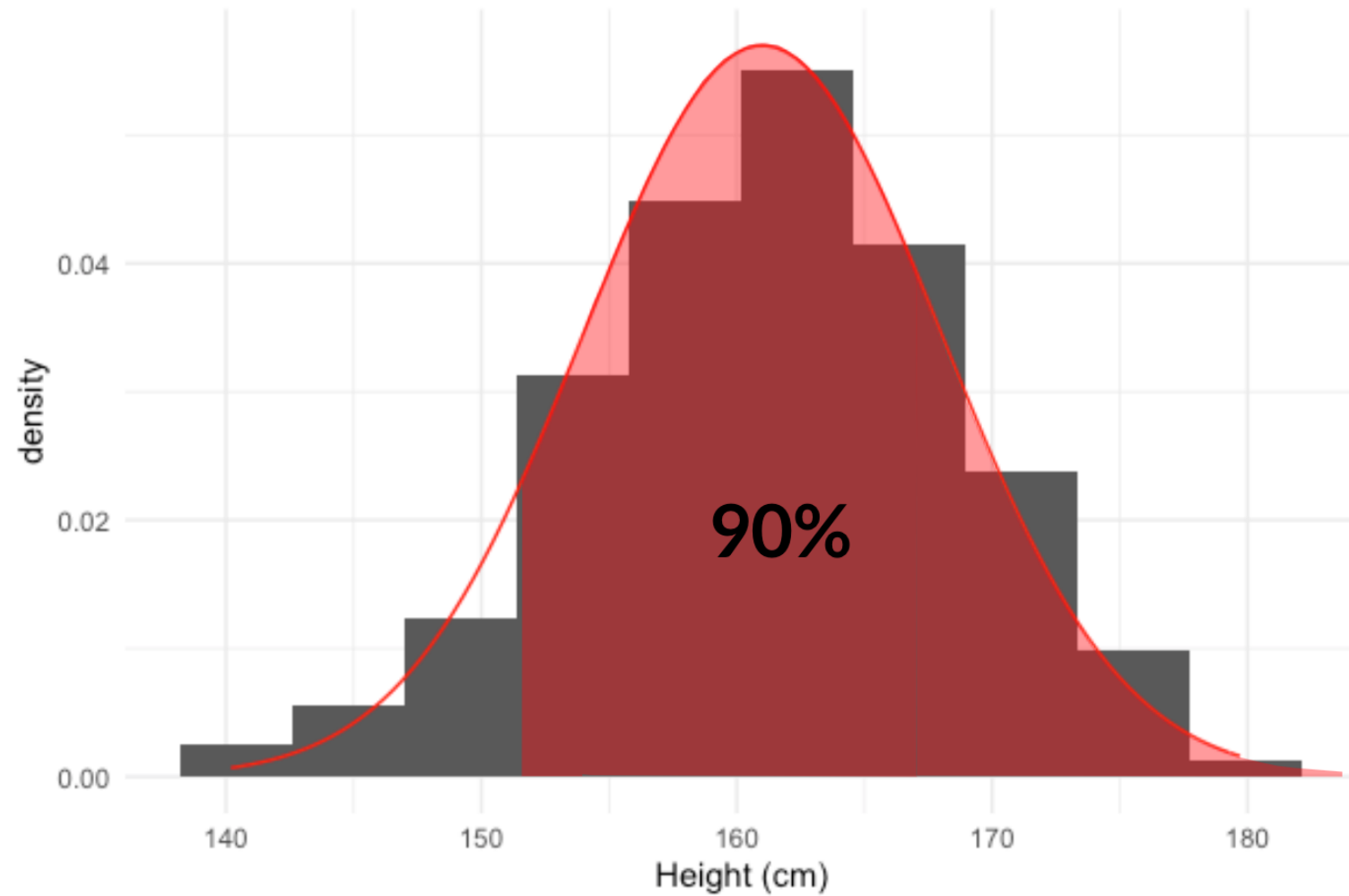
# What height are 90% of women shorter than?



```
norm.ppf(0.9, 161, 7)
```

169.97086

# What height are 90% of women taller than?



```
norm.ppf((1-0.9), 161, 7)
```

152.029

# Generating random numbers

```
# Generate 10 random heights  
norm.rvs(161, 7, size=10)
```

```
array([155.5758223 , 155.13133235, 160.06377097, 168.33345778,  
       165.92273375, 163.32677057, 165.13280753, 146.36133538,  
       149.07845021, 160.5790856  ])
```

# Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON

# The central limit theorem

INTRODUCTION TO STATISTICS IN PYTHON



**Maggie Matsui**  
Content Developer, DataCamp

# Rolling the dice 5 times

```
die = pd.Series([1, 2, 3, 4, 5, 6])  
# Roll 5 times  
samp_5 = die.sample(5, replace=True)  
print(samp_5)
```

```
array([3, 1, 4, 1, 1])
```

```
np.mean(samp_5)
```

```
2.0
```



# Rolling the dice 5 times

```
# Roll 5 times and take mean  
samp_5 = die.sample(5, replace=True)  
np.mean(samp_5)
```

4.4

```
samp_5 = die.sample(5, replace=True)  
np.mean(samp_5)
```

3.8

# Rolling the dice 5 times 10 times

*Repeat 10 times:*

- Roll 5 times
- Take the mean

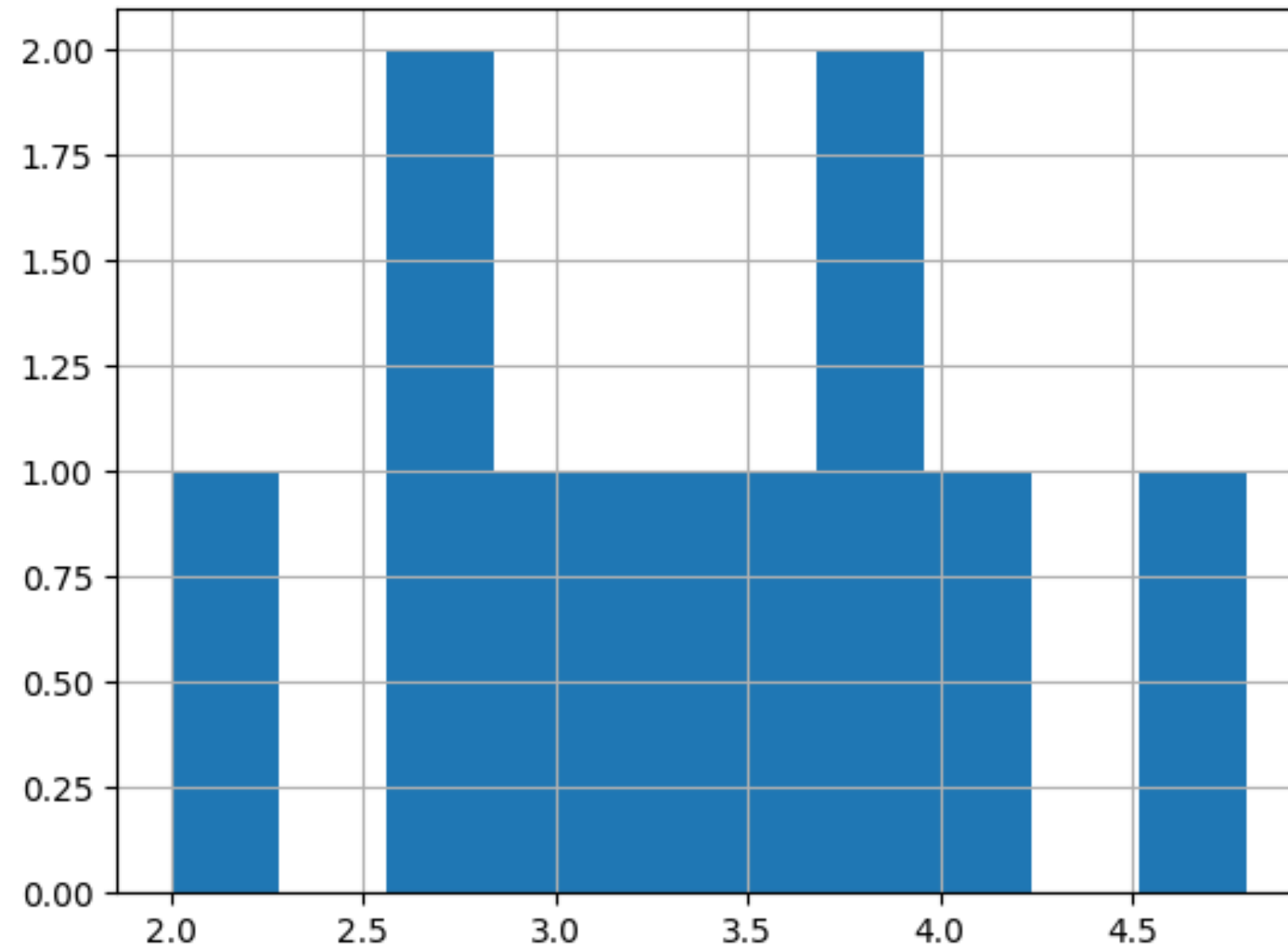
```
sample_means = []  
for i in range(10):  
    samp_5 = die.sample(5, replace=True)  
    sample_means.append(np.mean(samp_5))  
print(sample_means)
```

```
[3.8, 4.0, 3.8, 3.6, 3.2, 4.8, 2.6,  
3.0, 2.6, 2.0]
```



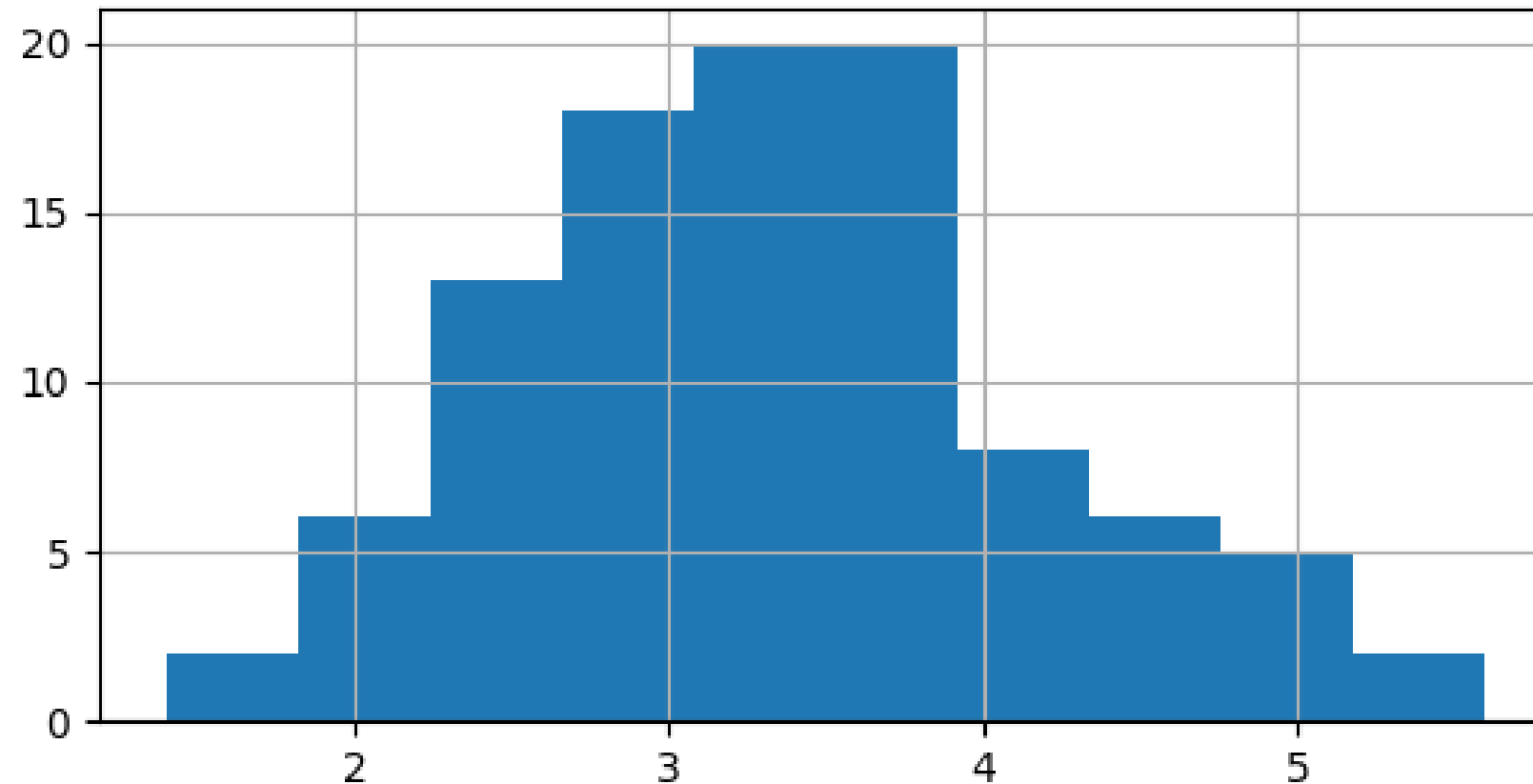
# Sampling distributions

*Sampling distribution of the sample mean*



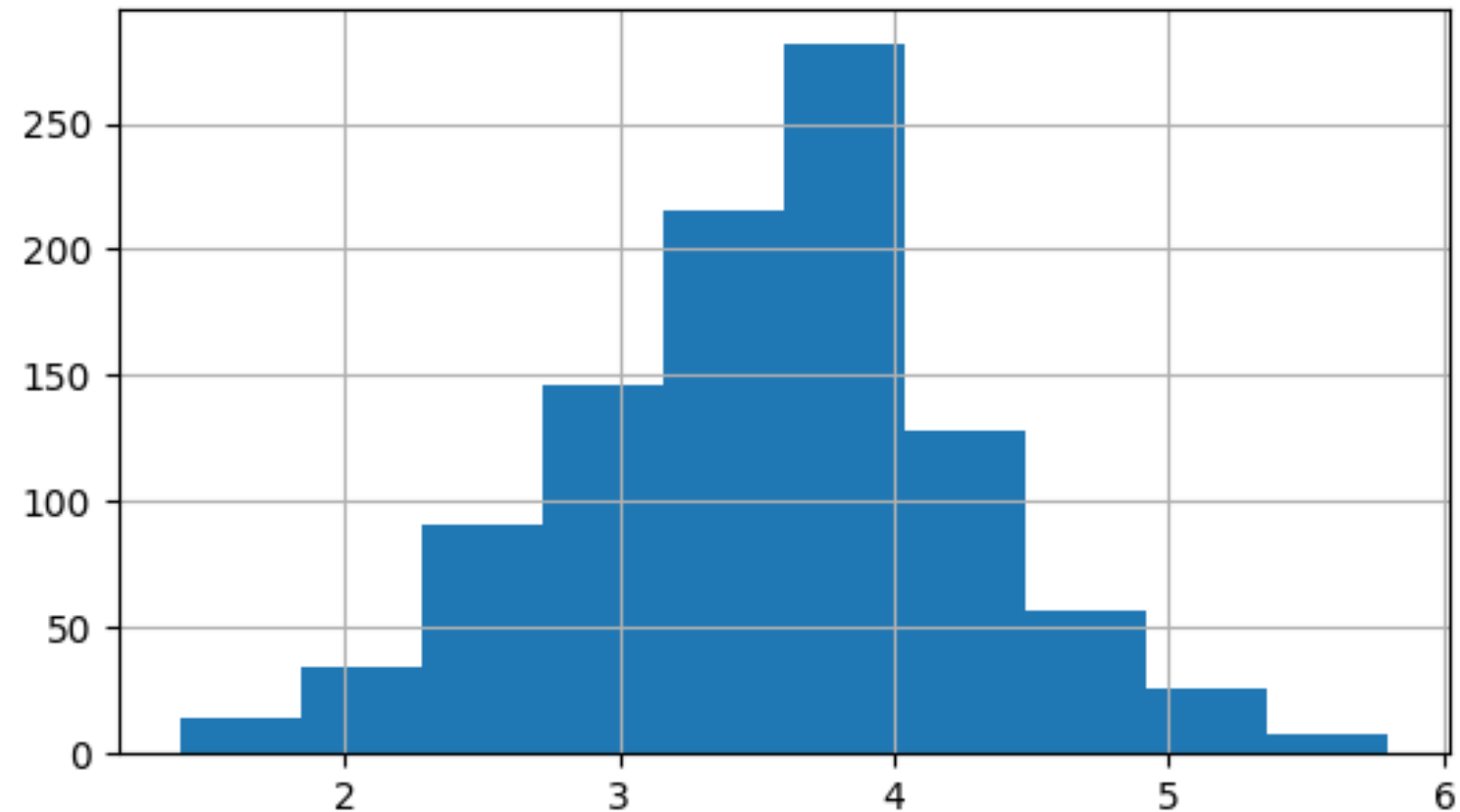
# 100 sample means

```
sample_means = []  
for i in range(100):  
    sample_means.append(np.mean(die.sample(5, replace=True)))
```



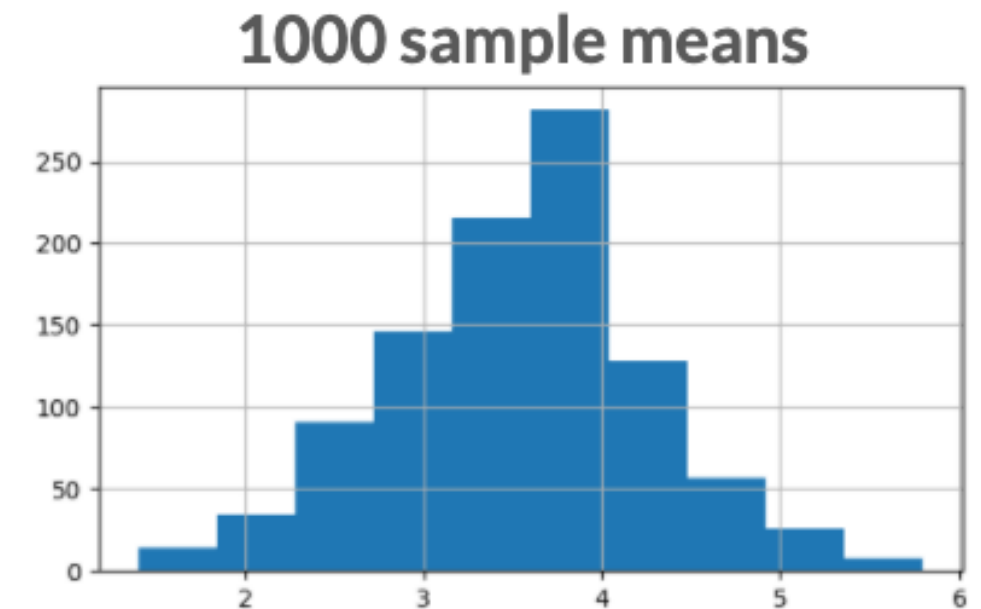
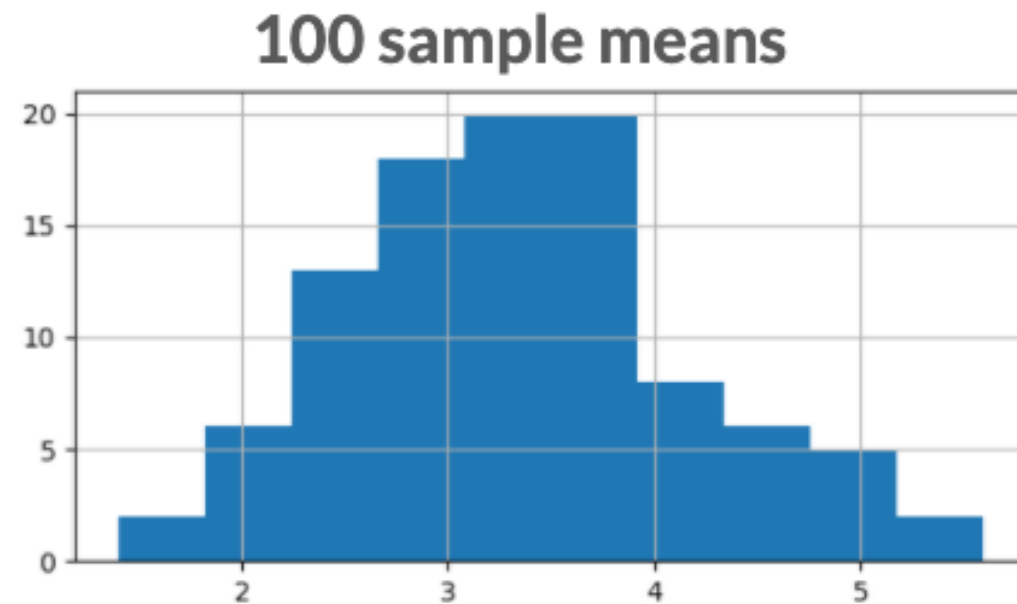
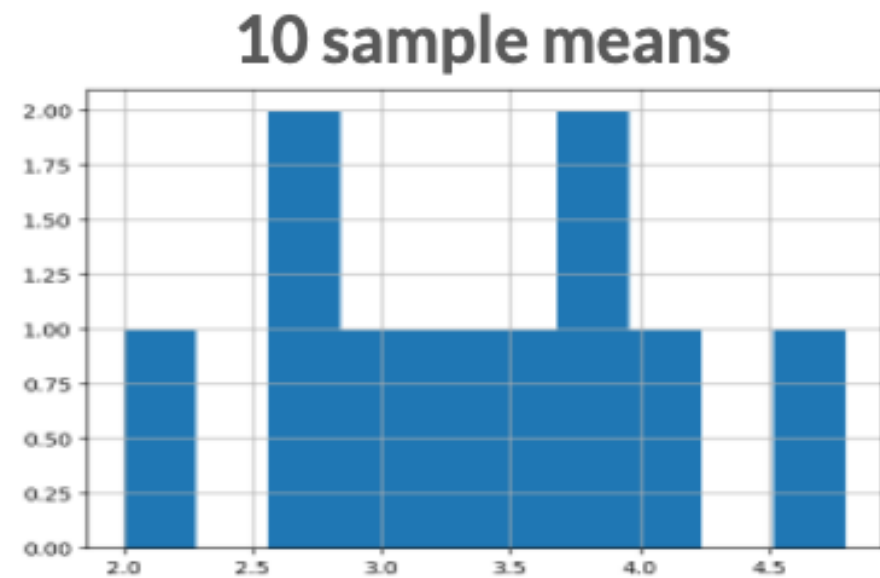
# 1000 sample means

```
sample_means = []  
for i in range(1000):  
    sample_means.append(np.mean(die.sample(5, replace=True)))
```



# Central limit theorem

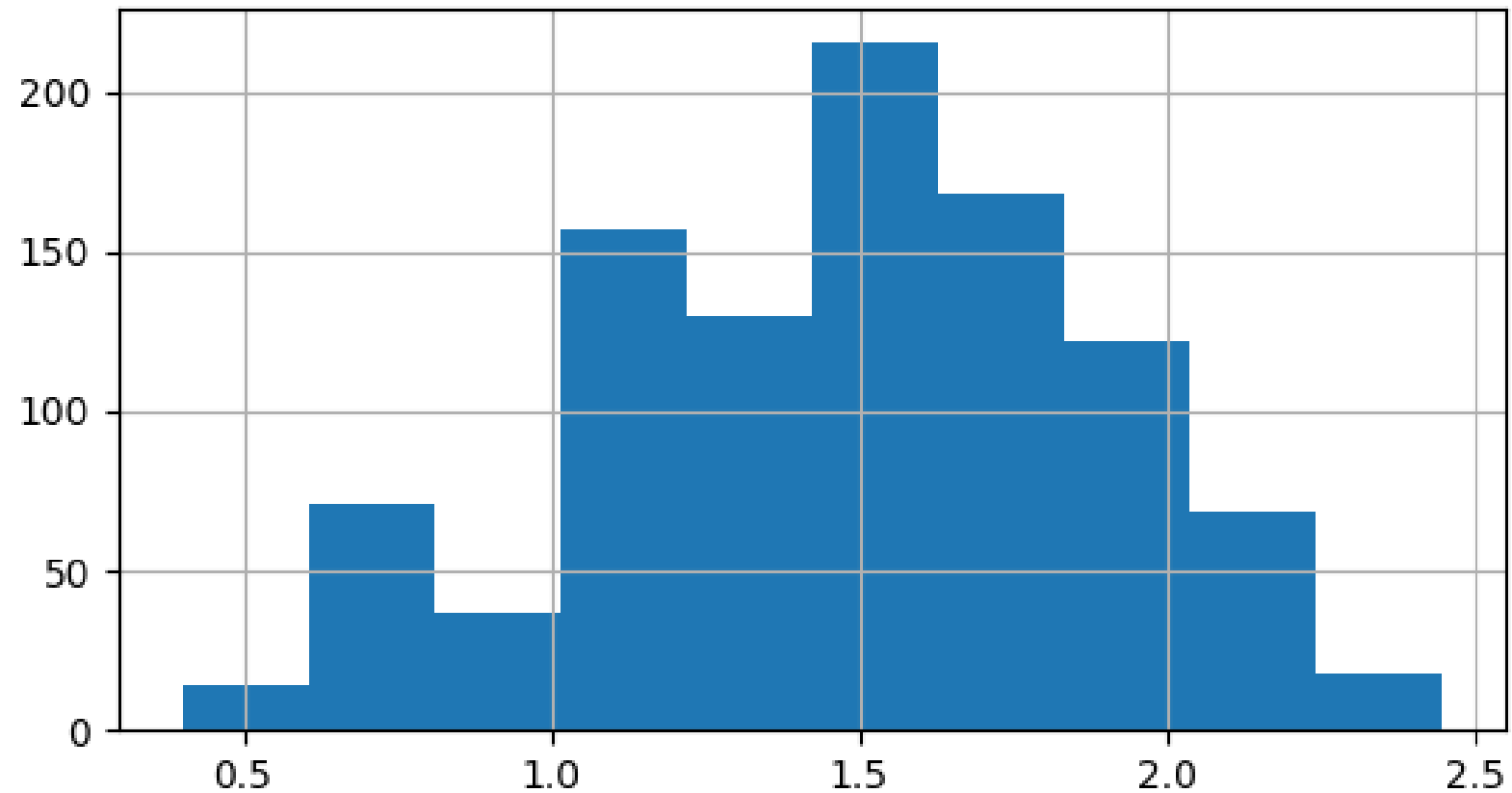
The sampling distribution of a statistic becomes closer to the normal distribution as the number of trials increases.



\* *Samples should be random and independent*

# Standard deviation and the CLT

```
sample_sds = []  
for i in range(1000):  
    sample_sds.append(np.std(die.sample(5, replace=True)))
```



# Proportions and the CLT

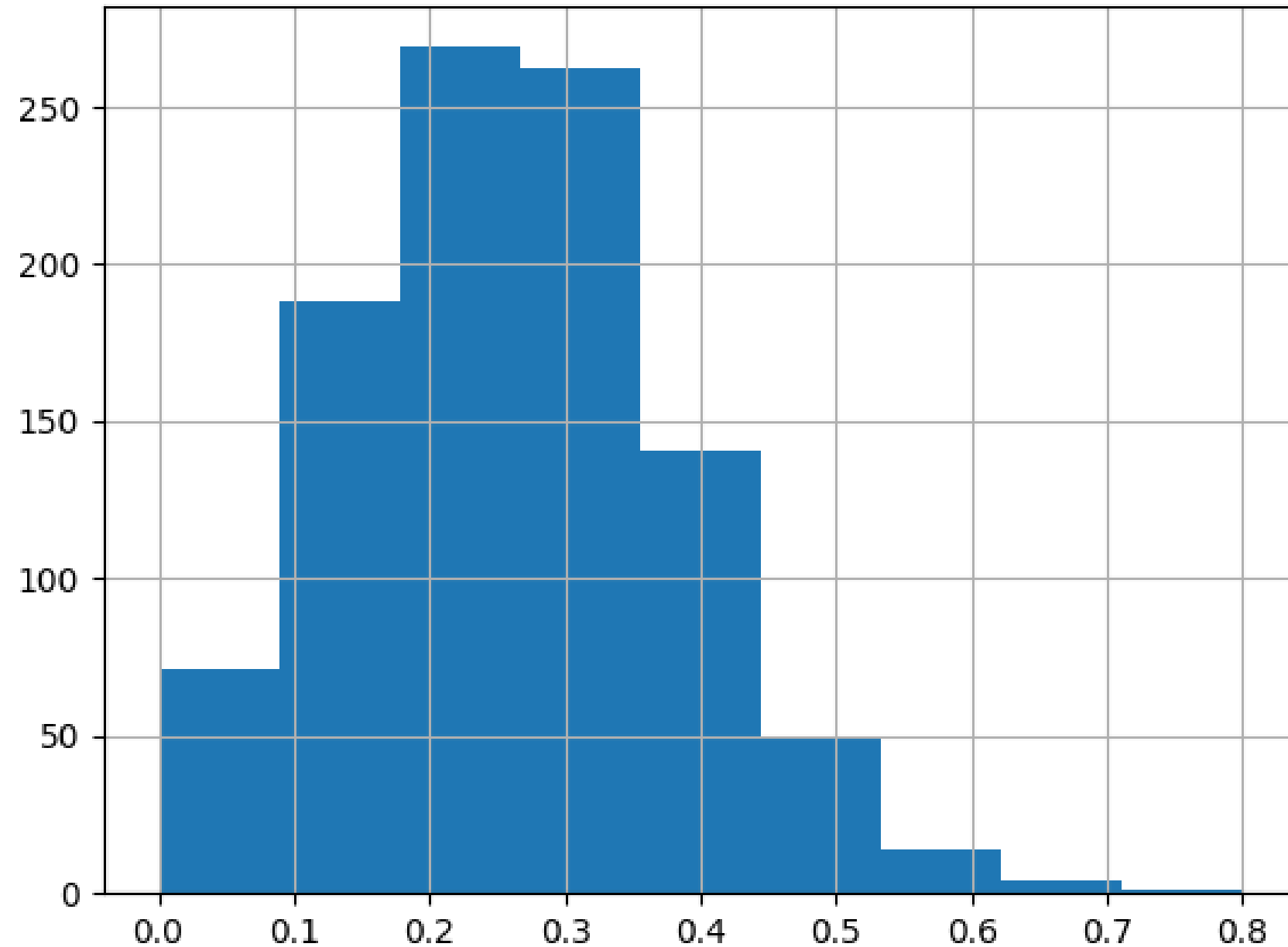
```
sales_team = pd.Series(["Amir", "Brian", "Claire", "Damian"])  
sales_team.sample(10, replace=True)
```

```
array(['Claire', 'Damian', 'Brian', 'Damian', 'Damian', 'Amir', 'Amir', 'Amir',  
      'Amir', 'Damian'], dtype=object)
```

```
sales_team.sample(10, replace=True)
```

```
array(['Brian', 'Amir', 'Brian', 'Claire', 'Brian', 'Damian', 'Claire', 'Brian',  
      'Claire', 'Claire'], dtype=object)
```

# Sampling distribution of proportion



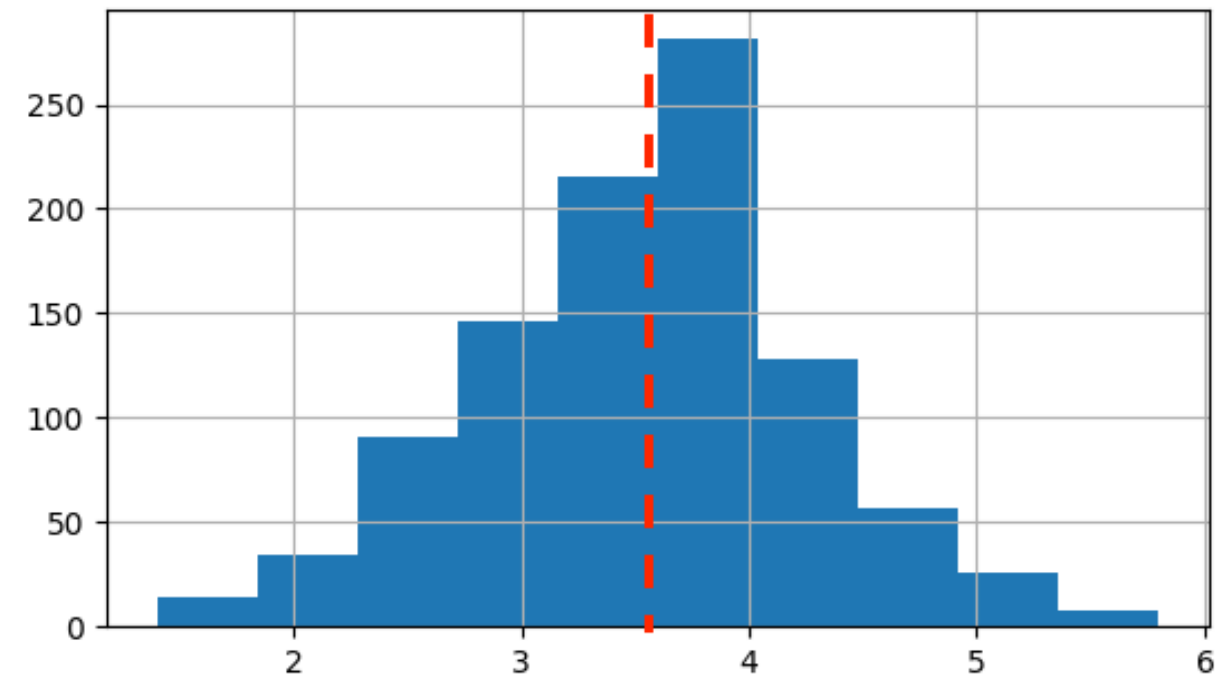
# Mean of sampling distribution

```
# Estimate expected value of die  
np.mean(sample_means)
```

3.48

```
# Estimate proportion of "Claire"s  
np.mean(sample_props)
```

0.26



- Estimate characteristics of unknown underlying distribution
- More easily estimate characteristics of large populations

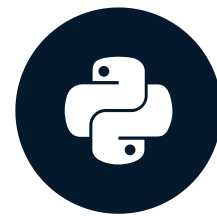


# Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON

# The Poisson distribution

INTRODUCTION TO STATISTICS IN PYTHON



**Maggie Matsui**  
Content Developer, DataCamp

# Poisson processes

- Events appear to happen at a certain rate, but completely at random
- Examples
  - Number of animals adopted from an animal shelter per week
  - Number of people arriving at a restaurant per hour
  - Number of earthquakes in California per year
- Time unit is irrelevant, as long as you use the same unit when talking about the same situation

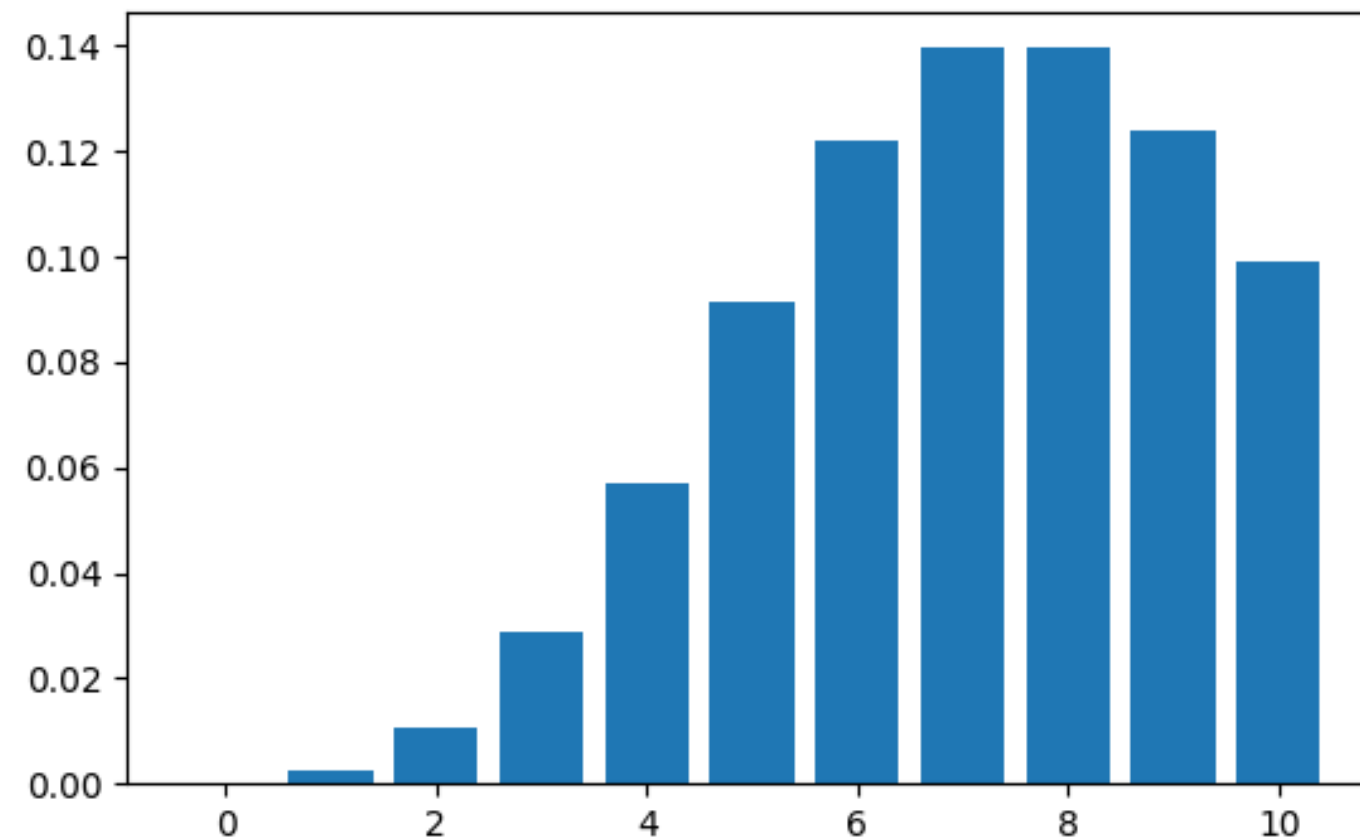


# Poisson distribution

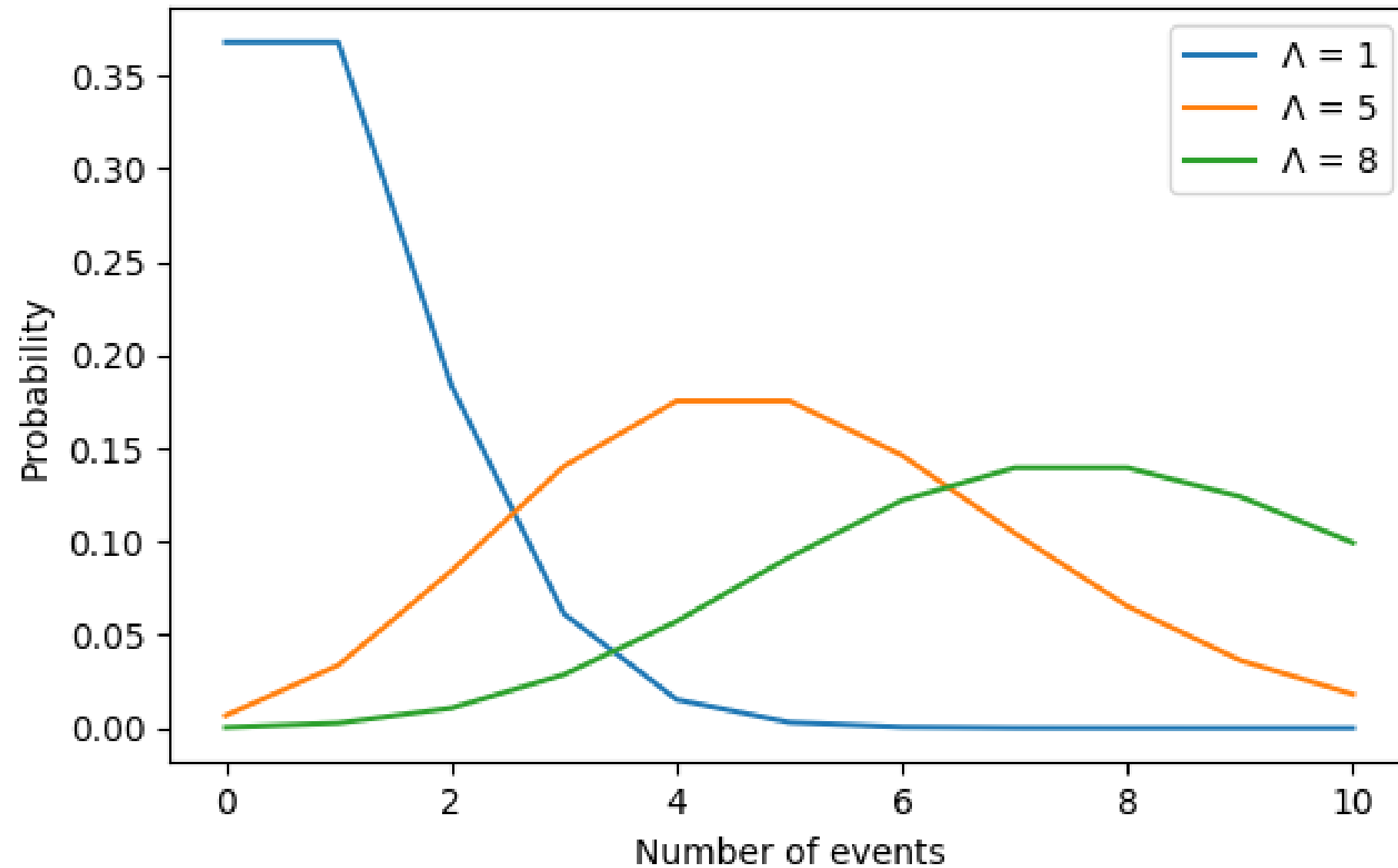
- Probability of some # of events occurring over a fixed period of time
- Examples
  - Probability of  $\geq 5$  animals adopted from an animal shelter per week
  - Probability of 12 people arriving at a restaurant per hour
  - Probability of  $< 20$  earthquakes in California per year

# Lambda ( $\lambda$ )

- $\lambda$  = average number of events per time interval
  - Average number of adoptions per week = 8



# Lambda is the distribution's peak



# Probability of a single value

If the average number of adoptions per week is 8, what is  $P(\# \text{ adoptions in a week} = 5)$ ?

```
from scipy.stats import poisson  
poisson.pmf(5, 8)
```

```
0.09160366
```

# Probability of less than or equal to

If the average number of adoptions per week is 8, what is  $P(\# \text{ adoptions in a week} \leq 5)$ ?

```
from scipy.stats import poisson  
poisson.cdf(5, 8)
```

```
0.1912361
```



# Probability of greater than

If the average number of adoptions per week is 8, what is  $P(\# \text{ adoptions in a week} > 5)$ ?

```
1 - poisson.cdf(5, 8)
```

```
0.8087639
```

If the average number of adoptions per week is 10, what is  $P(\# \text{ adoptions in a week} > 5)$ ?

```
1 - poisson.cdf(5, 10)
```

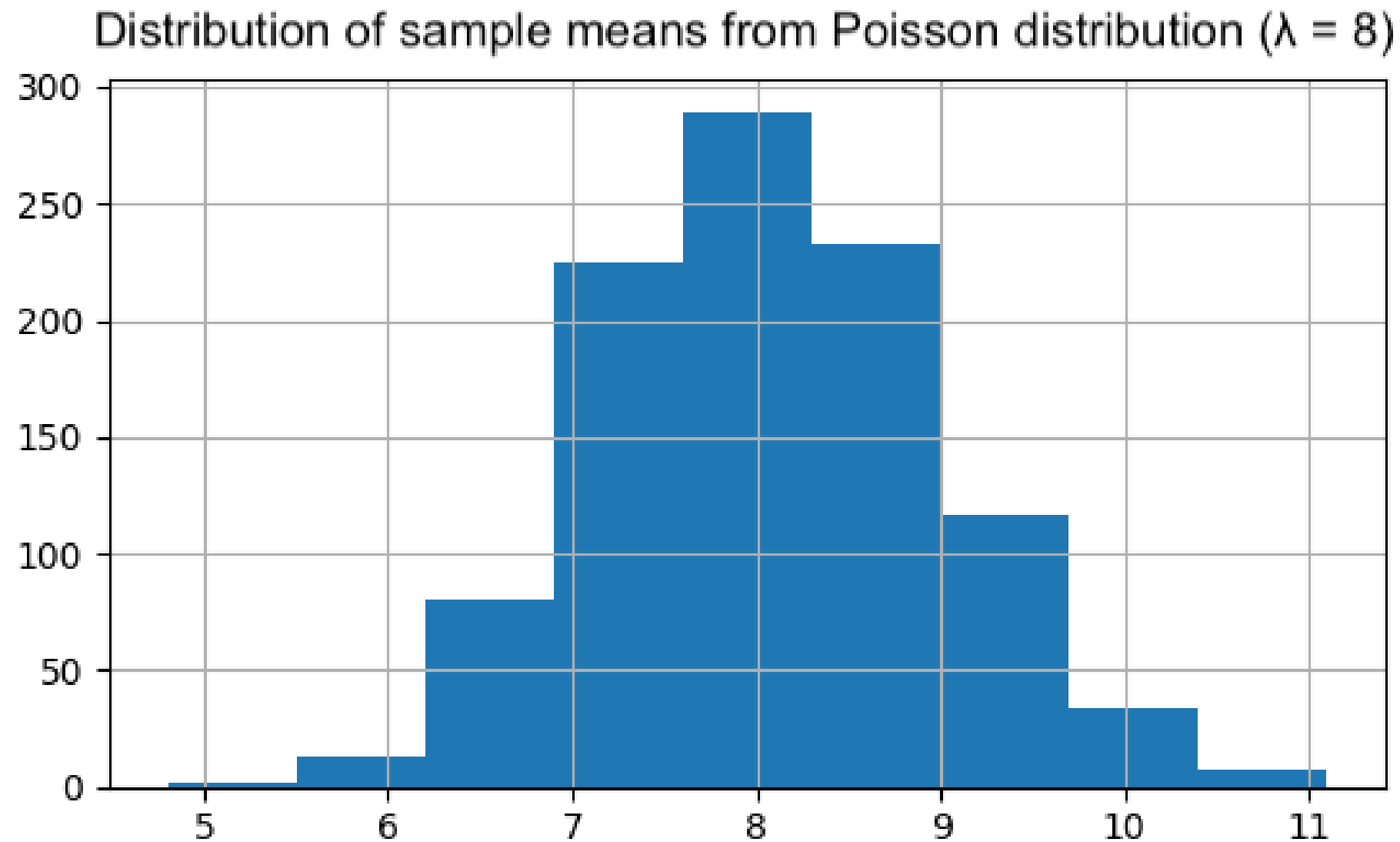
```
0.932914
```

# Sampling from a Poisson distribution

```
from scipy.stats import poisson  
poisson.rvs(8, size=10)
```

```
array([ 9,  9,  8,  7, 11,  3, 10,  6,  8, 14])
```

# The CLT still applies!



# Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON

# More probability distributions

INTRODUCTION TO STATISTICS IN PYTHON



**Maggie Matsui**

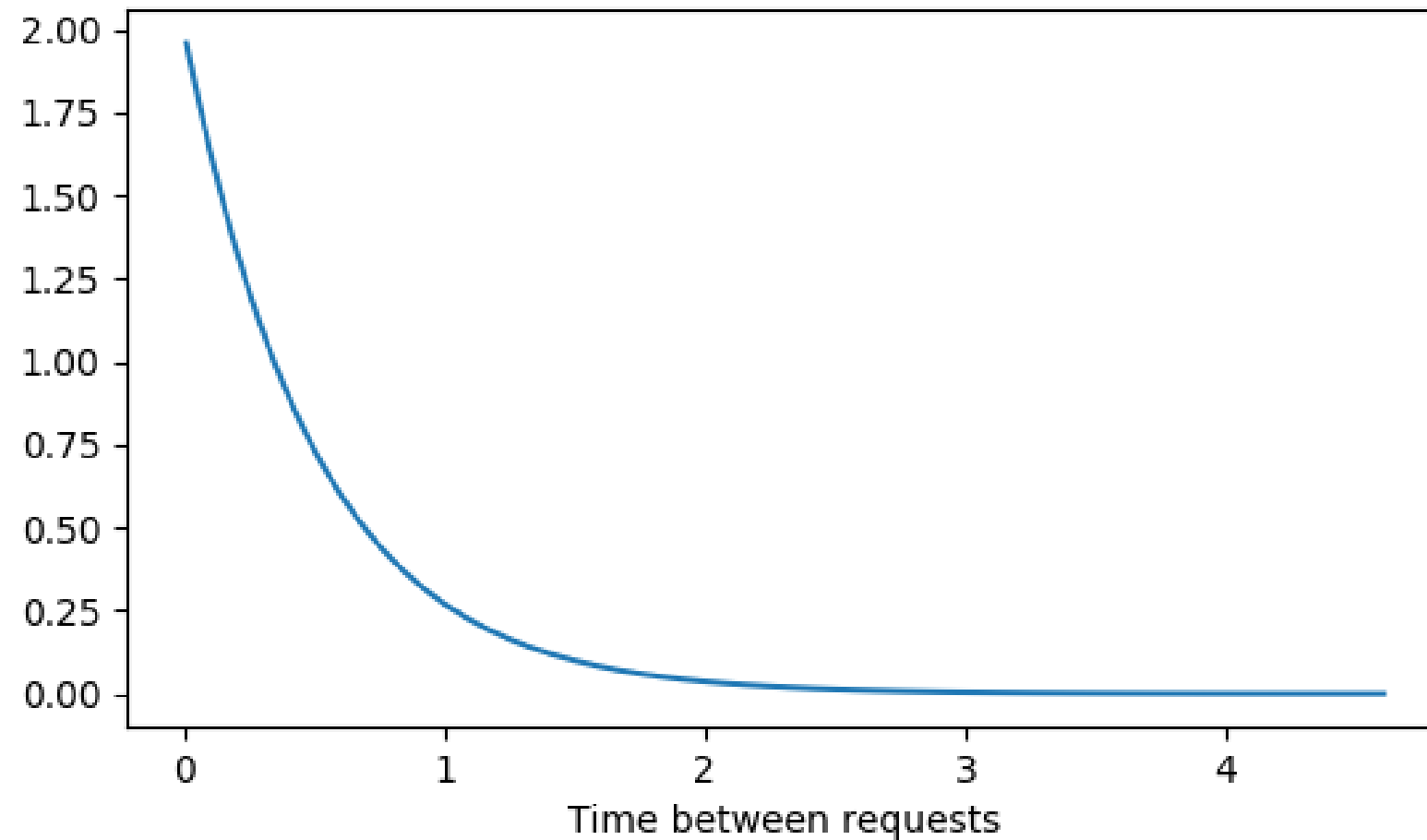
Content Developer, DataCamp

# Exponential distribution

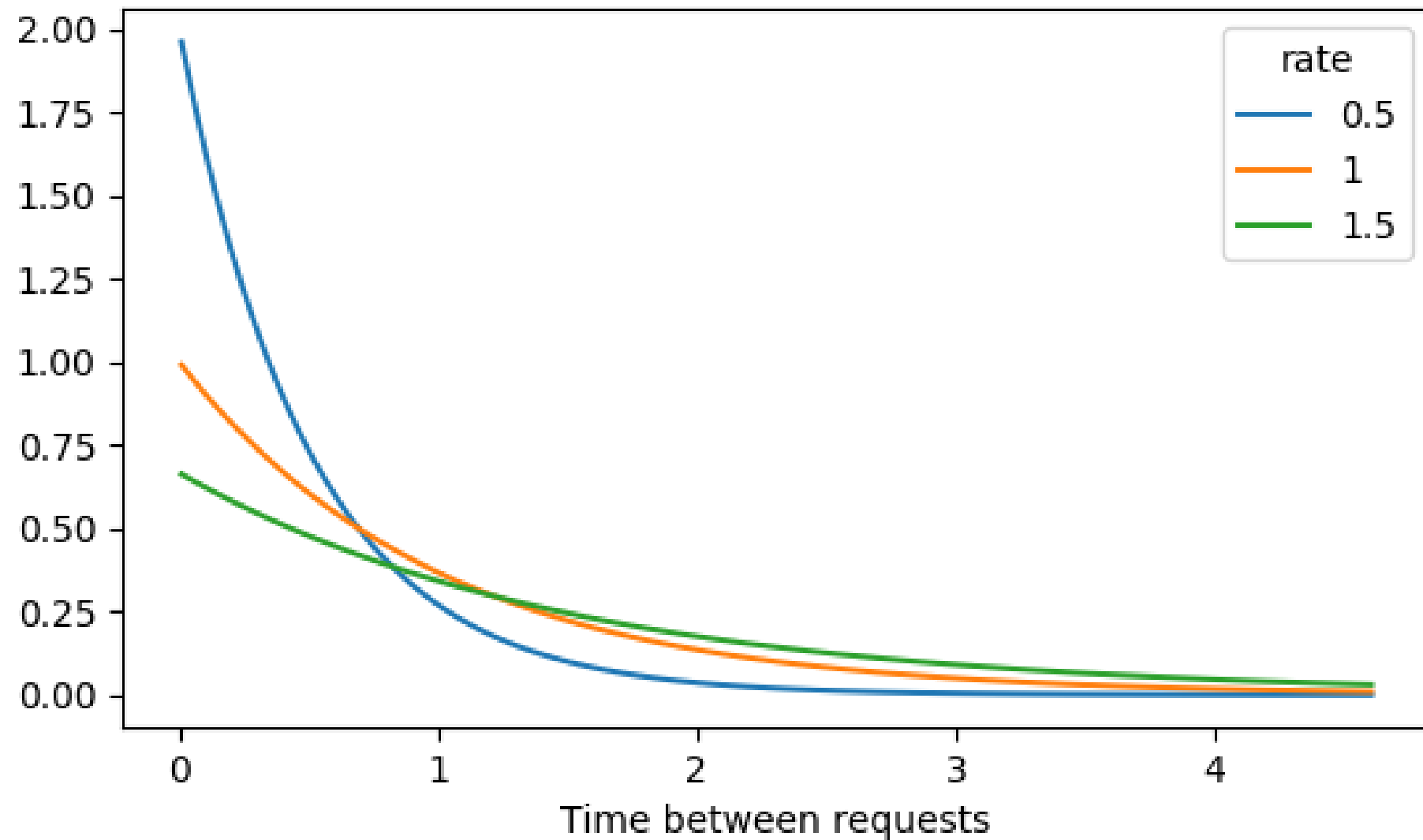
- Probability of time between Poisson events
- Examples
  - Probability of  $> 1$  day between adoptions
  - Probability of  $< 10$  minutes between restaurant arrivals
  - Probability of 6-8 months between earthquakes
- Also uses lambda (rate)
- Continuous (time)

# Customer service requests

- On average, one customer service ticket is created every 2 minutes
  - $\lambda = 0.5$  customer service tickets created each minute



# Lambda in exponential distribution





# Expected value of exponential distribution

In terms of rate (Poisson):

- $\lambda = 0.5$  requests per minute

In terms of time between events (exponential):

- $1/\lambda = 1$  request per 2 minutes
- $1/0.5 = 2$

# How long until a new request is created?

$$P(\text{wait} < 1 \text{ min}) =$$

```
from scipy.stats import expon
```

```
expon.cdf(1, scale=2)
```

- `scale` =  $1/\lambda = 1/0.5 = 2$

```
0.3934693402873666
```

$$P(\text{wait} > 4 \text{ min}) =$$

```
1 - expon.cdf(4, scale=2)
```

```
0.1353352832366127
```

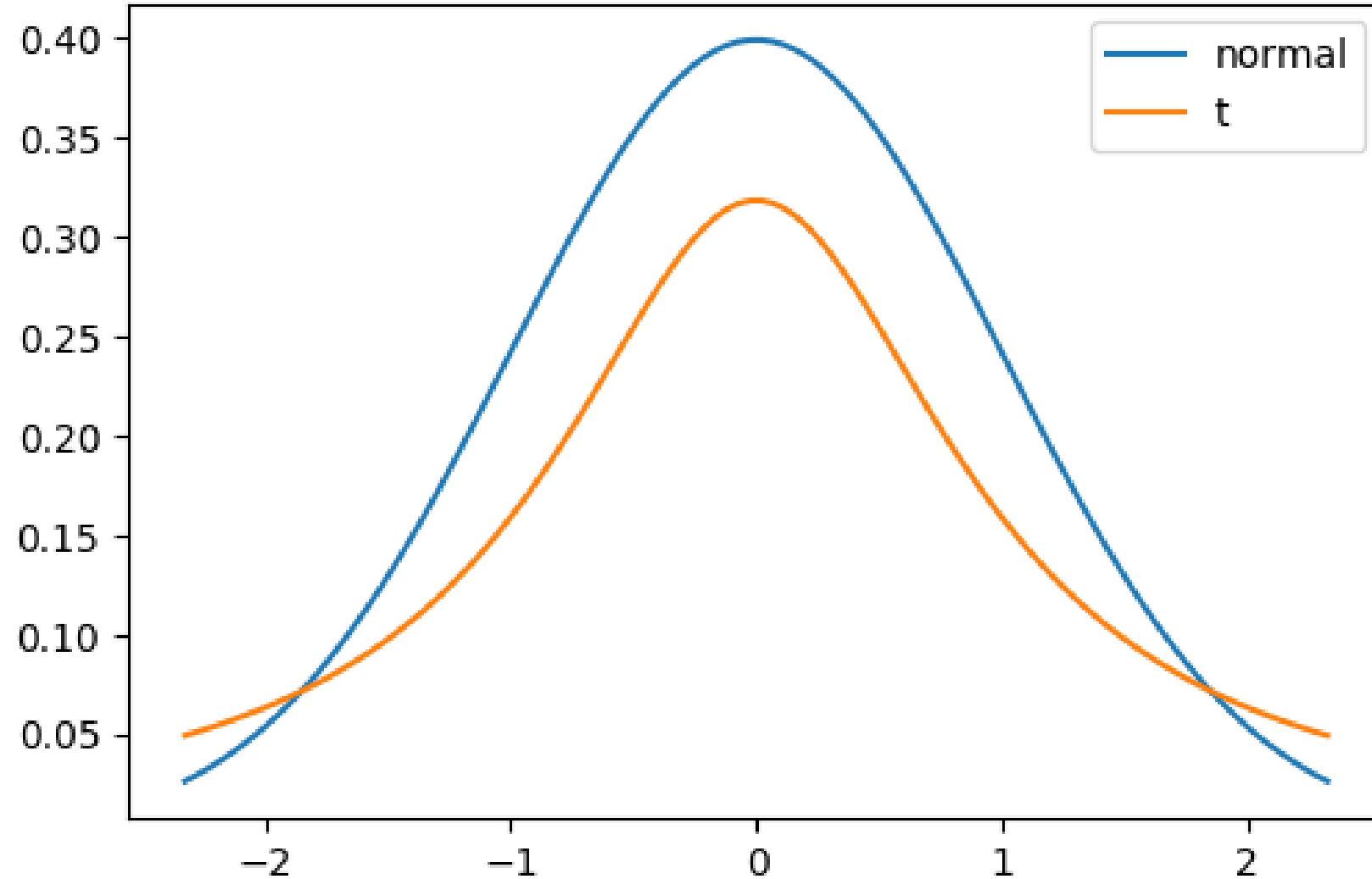
$$P(1 \text{ min} < \text{wait} < 4 \text{ min}) =$$

```
expon.cdf(4, scale=2) - expon.cdf(1, scale=2)
```

```
0.4711953764760207
```

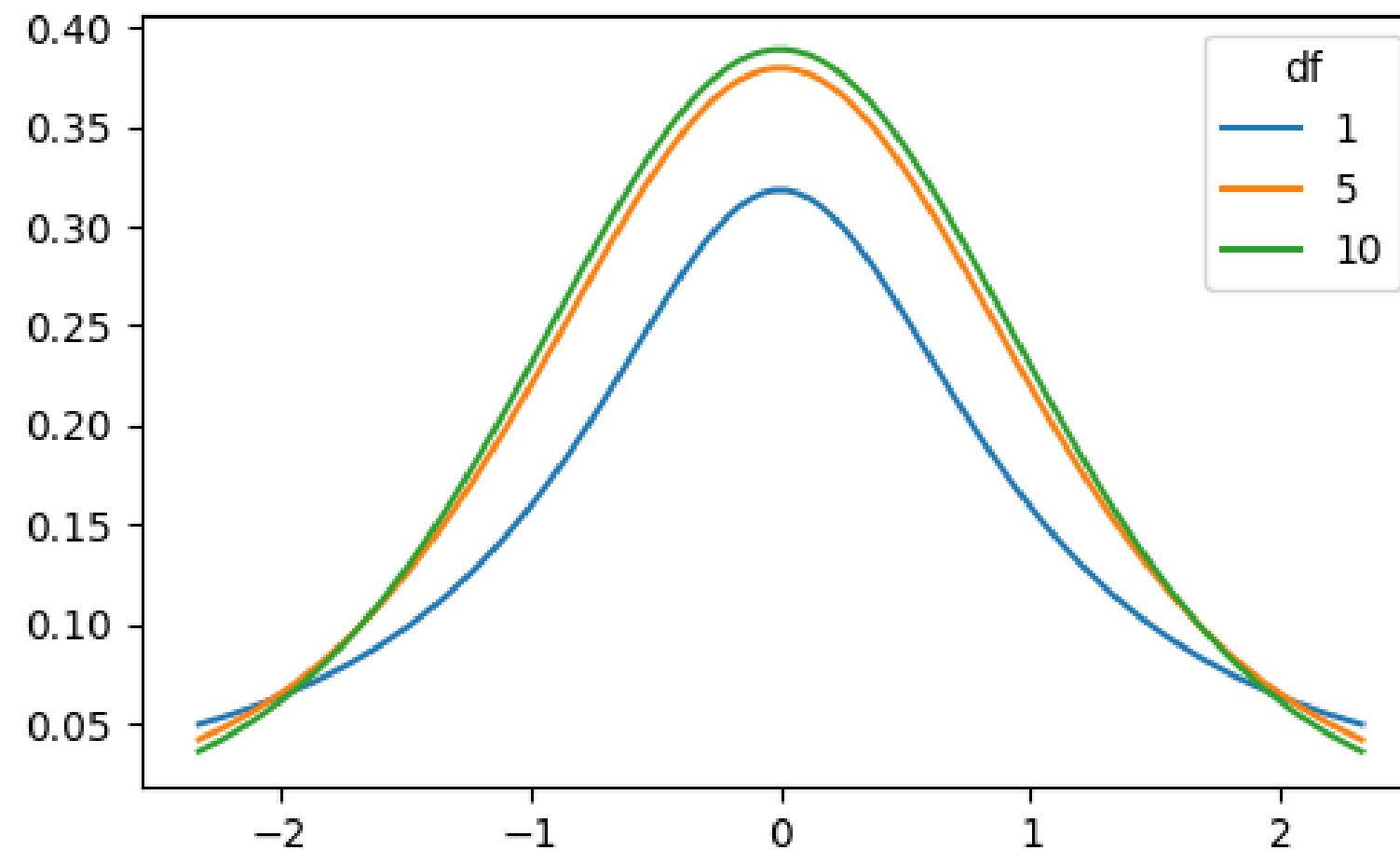
# (Student's) t-distribution

- Similar shape as the normal distribution



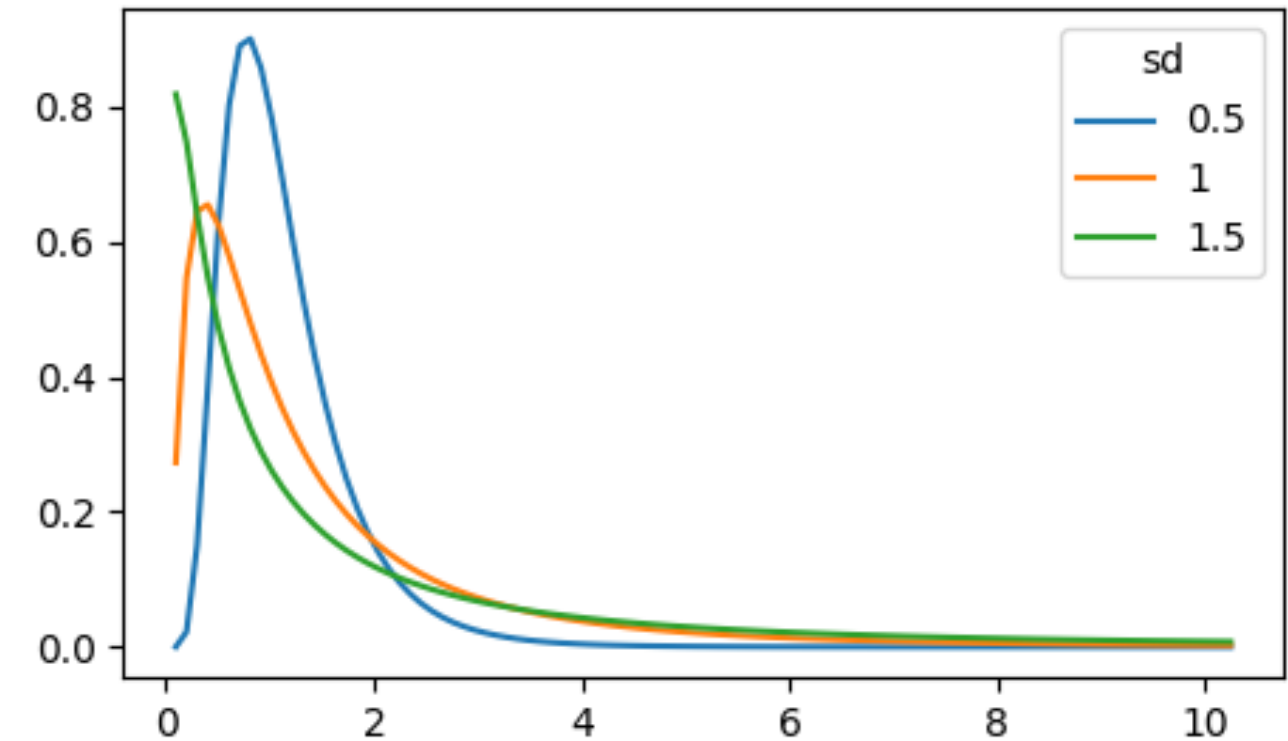
# Degrees of freedom

- Has parameter degrees of freedom (df) which affects the thickness of the tails
  - Lower df = thicker tails, higher standard deviation
  - Higher df = closer to normal distribution



# Log-normal distribution

- Variable whose logarithm is normally distributed
- Examples:
  - Length of chess games
  - Adult blood pressure
  - Number of hospitalizations in the 2003 SARS outbreak



# Let's practice!

INTRODUCTION TO STATISTICS IN PYTHON