

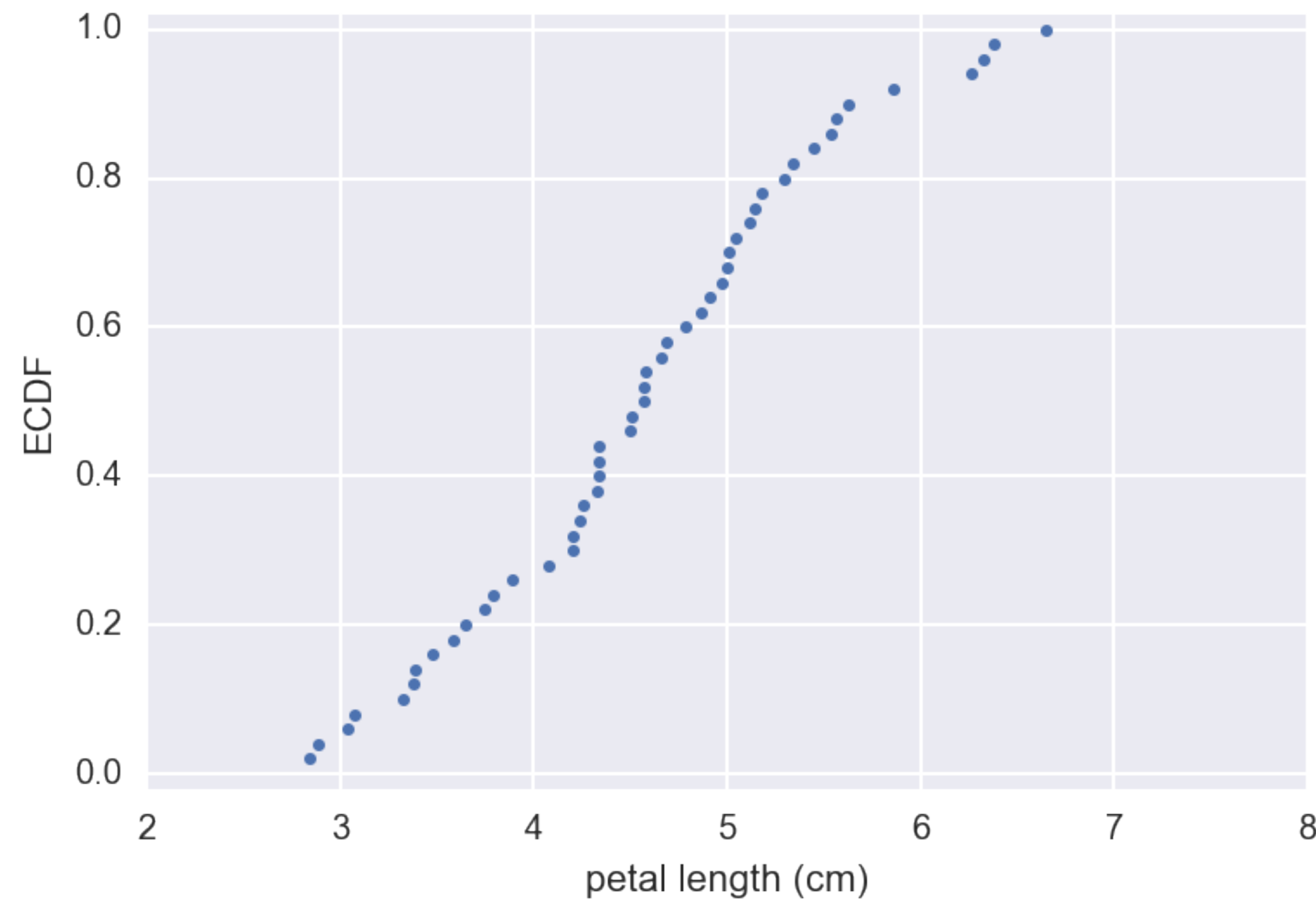


STATISTICAL THINKING IN PYTHON I

Probabilistic logic and statistical inference

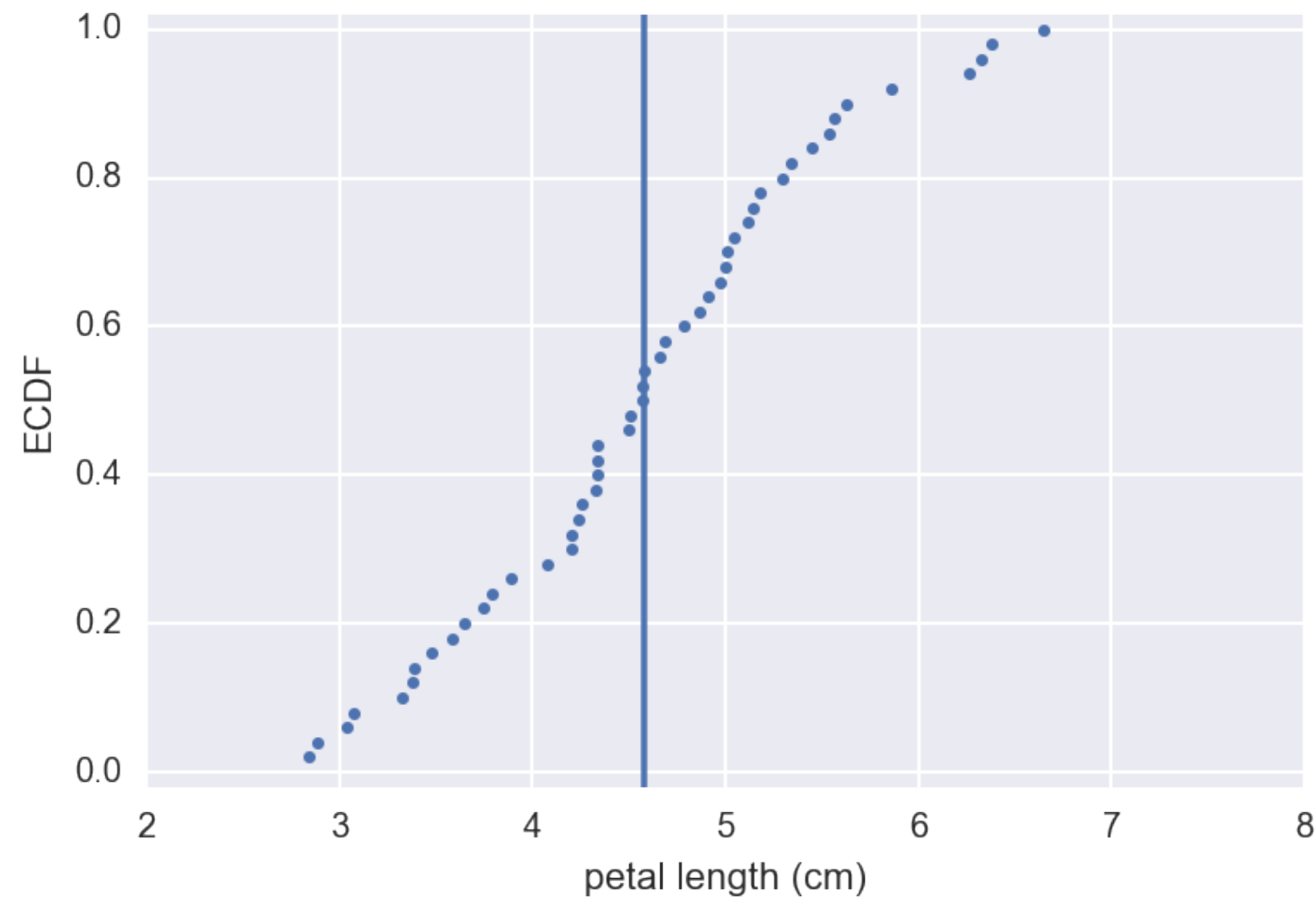


50 measurements of petal length

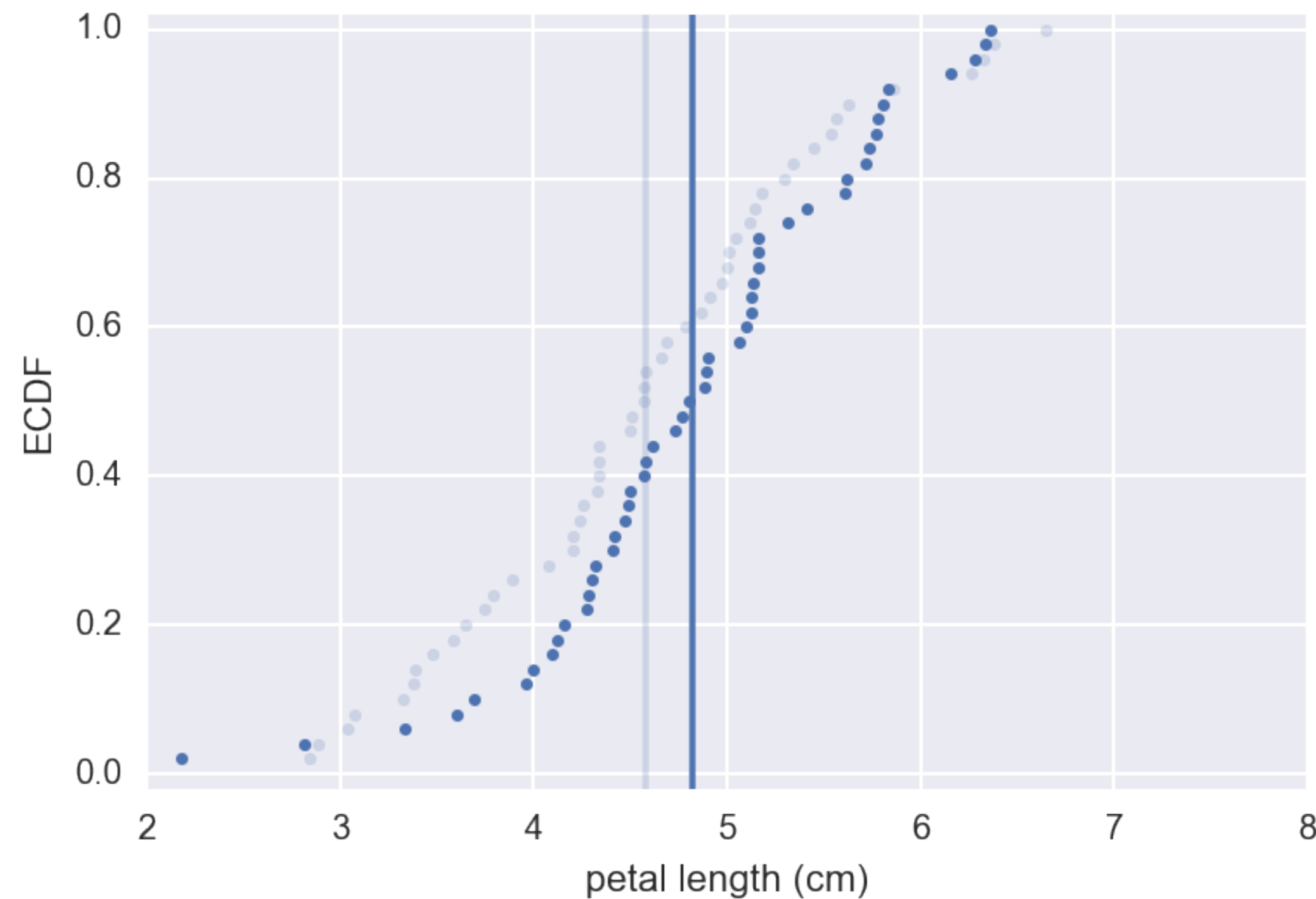




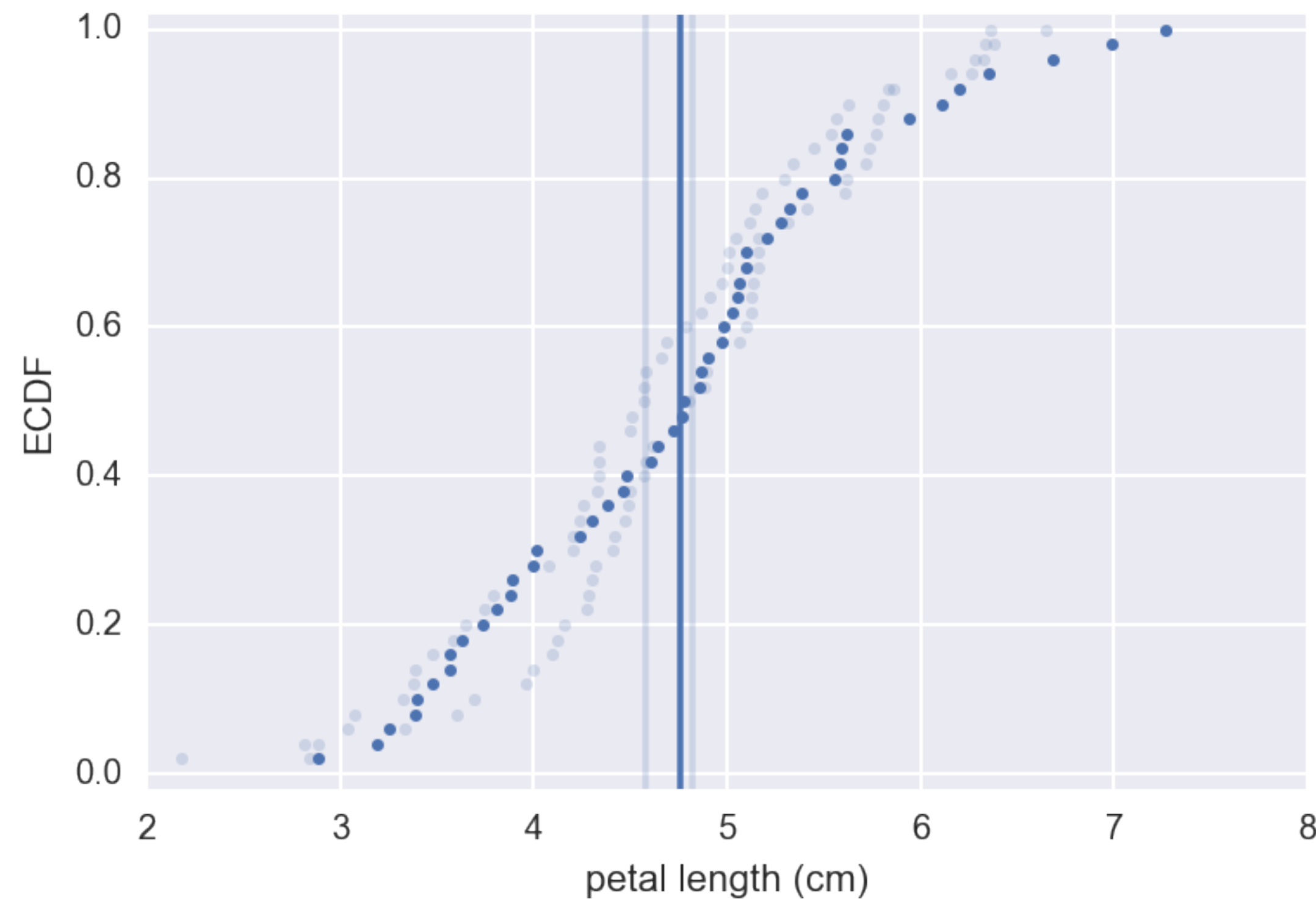
50 measurements of petal length



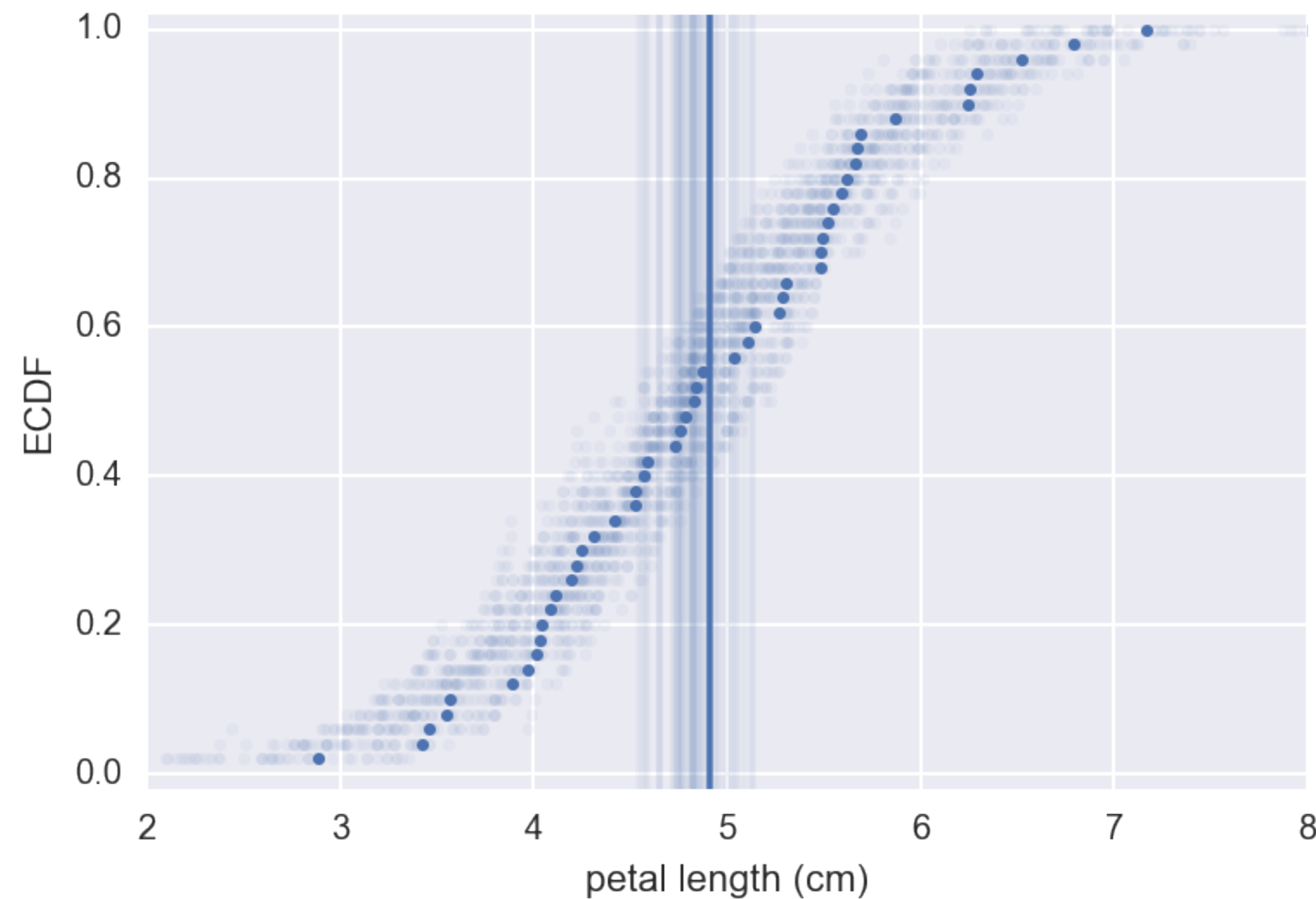
50 measurements of petal length



50 measurements of petal length

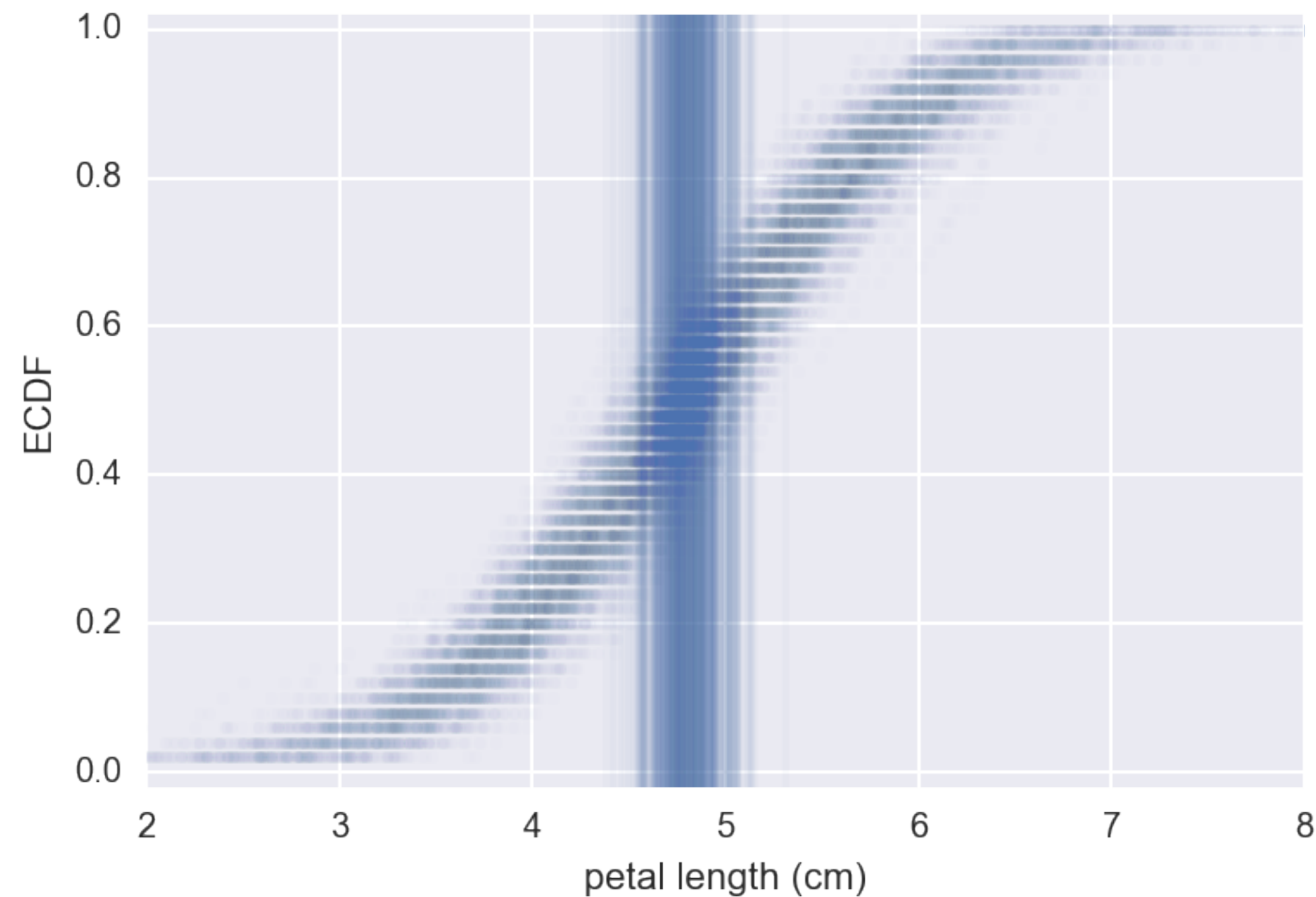


50 measurements of petal length





Repeats of 50 measurements of petal length





STATISTICAL THINKING IN PYTHON I

Let's practice!



STATISTICAL THINKING IN PYTHON I

Random number generators and hacker statistics



Hacker statistics

- Uses simulated repeated measurements to compute probabilities.



Blaise Pascal



The `np.random` module

- Suite of functions based on *random number generation*
- `np.random.random()`:
draw a number between 0 and 1



The `np.random` module

- Suite of functions based on *random number generation*
- `np.random.random()`:
draw a number between 0 and 1

< 0.5



≥ 0.5



Bernoulli trial

- An experiment that has two options, "success" (True) and "failure" (False).

Random number seed

- Integer fed into random number generating algorithm
- Manually seed random number generator if you need reproducibility
- Specified using `np.random.seed()`



Simulating 4 coin flips

```
In [1]: import numpy as np
```

```
In [2]: np.random.seed(42)
```

```
In [3]: random_numbers = np.random.random(size=4)
```

```
In [4]: random_numbers
```

```
Out[4]: array([ 0.37454012,  0.95071431,  0.73199394,  
               0.59865848])
```

```
In [5]: heads = random_numbers < 0.5
```

```
In [6]: heads
```

```
Out[6]: array([ True, False, False, False], dtype=bool)
```

```
In [7]: np.sum(heads)
```

```
Out[7]: 1
```



Simulating 4 coin flips

```
In [1]: n_all_heads = 0 # Initialize number of 4-heads trials
```

```
In [2]: for _ in range(10000):  
...:     heads = np.random.random(size=4) < 0.5  
...:     n_heads = np.sum(heads)  
...:     if n_heads == 4:  
...:         n_all_heads += 1  
...:  
...:
```

```
In [3]: n_all_heads / 10000
```

```
Out[3]: 0.0621
```

Hacker stats probabilities

- Determine how to simulate data
- Simulate many many times
- Probability is approximately fraction of trials with the outcome of interest



STATISTICAL THINKING IN PYTHON I

Let's practice!



STATISTICAL THINKING IN PYTHON I

Probability distributions and stories: The Binomial distribution

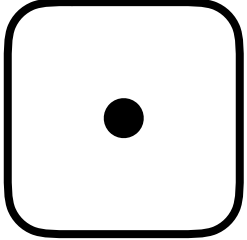
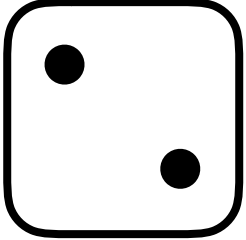
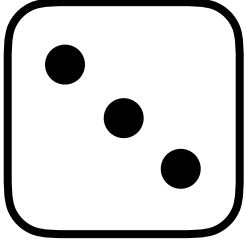
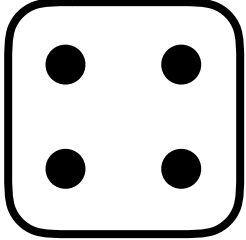
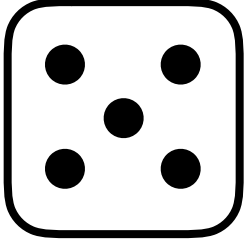
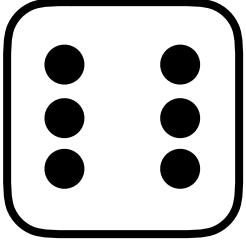
Probability mass function (PMF)

- The set of probabilities of discrete outcomes

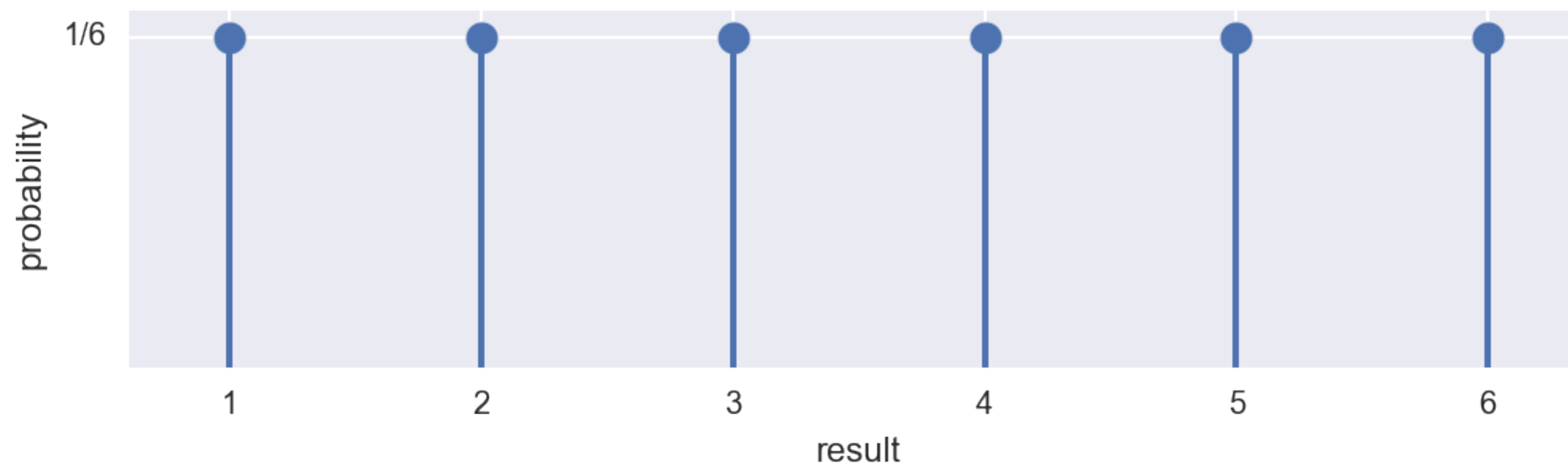


Discrete Uniform PMF

Tabular

					
$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$

Graphical





Probability distribution

- A mathematical description of outcomes

Discrete Uniform distribution: the story

- The outcome of rolling a single fair die is Discrete Uniformly distributed.



Binomial distribution: the story

- The number r of successes in n Bernoulli trials with probability p of success, is Binomially distributed
- The number r of heads in 4 coin flips with probability 0.5 of heads, is Binomially distributed



Sampling from the Binomial distribution

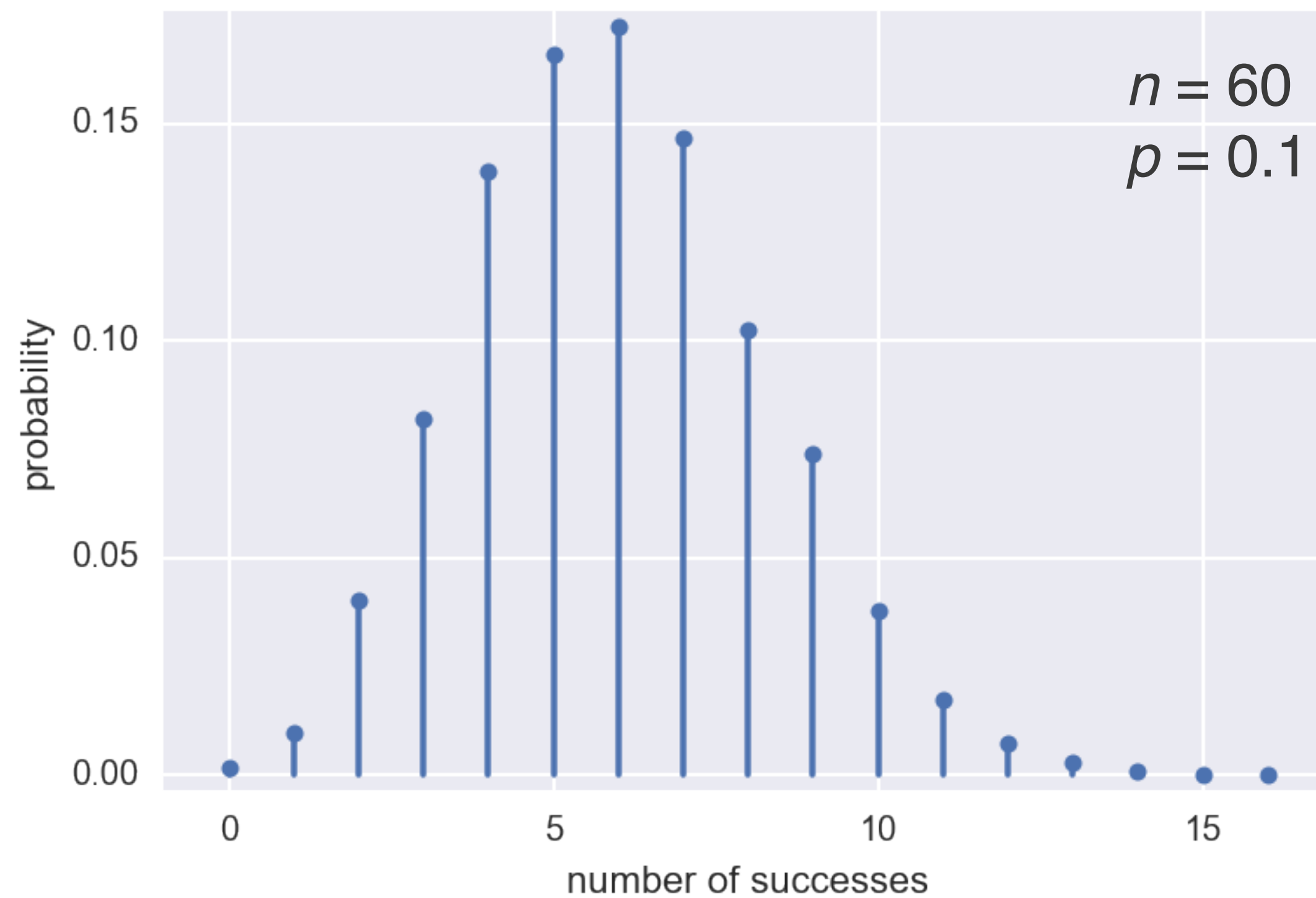
```
In [1]: np.random.binomial(4, 0.5)
Out[1]: 2
```

```
In [2]: np.random.binomial(4, 0.5, size=10)
Out[2]: array([4, 3, 2, 1, 1, 0, 3, 2, 3, 0])
```



The Binomial PMF

```
In [1]: samples = np.random.binomial(60, 0.1, size=10000)
```





The Binomial CDF

```
In [1]: import matplotlib.pyplot as plt
```

```
In [2]: import seaborn as sns
```

```
In [3]: sns.set()
```

```
In [4]: x, y = ecdf(samples)
```

```
In [5]: _ = plt.plot(x, y, marker='.', linestyle='none')
```

```
In [6]: plt.margins(0.02)
```

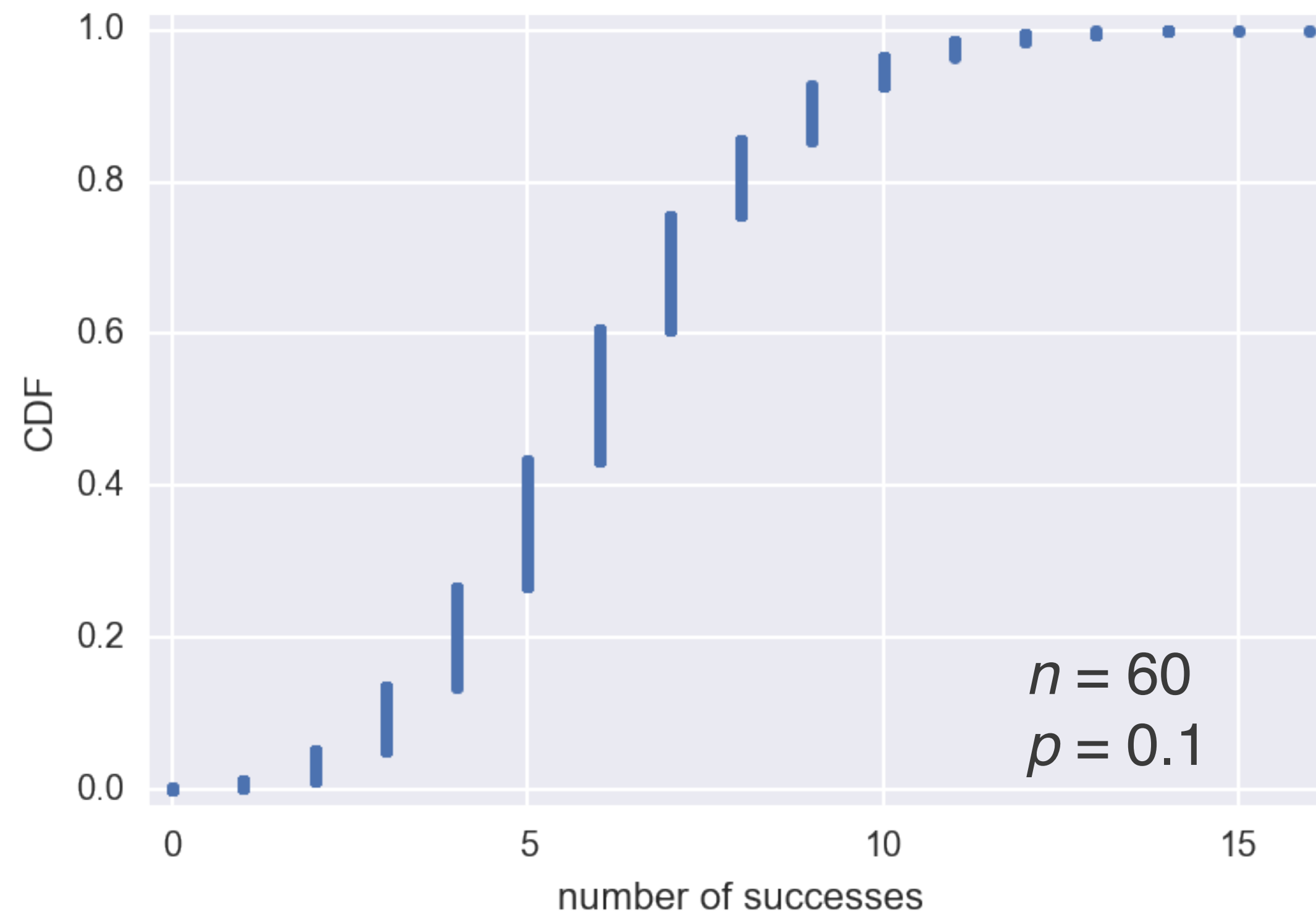
```
In [7]: _ = plt.xlabel('number of successes')
```

```
In [8]: _ = plt.ylabel('CDF')
```

```
In [9]: plt.show()
```



The Binomial CDF





STATISTICAL THINKING IN PYTHON I

Let's practice!



STATISTICAL THINKING IN PYTHON I

Poisson processes and the Poisson distribution

Poisson process

- The timing of the next event is completely independent of when the previous event happened

Examples of Poisson processes

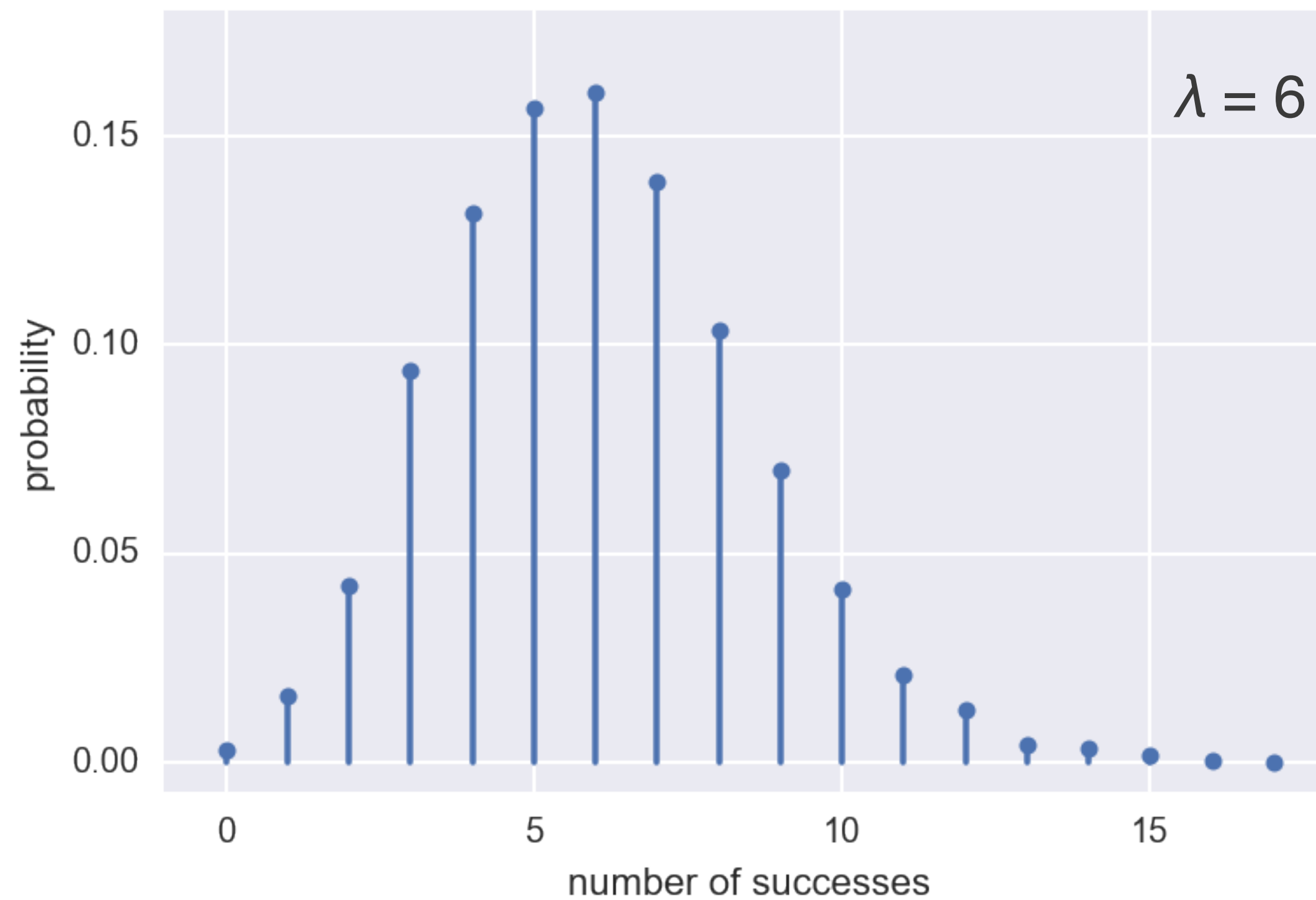
- Natural births in a given hospital
- Hit on a website during a given hour
- Meteor strikes
- Molecular collisions in a gas
- Aviation incidents
- Buses in Poissonville



Poisson distribution


- The number r of arrivals of a Poisson process in a given time interval with average rate of λ arrivals per interval is Poisson distributed.
- The number r of hits on a website in one hour with an average hit rate of 6 hits per hour is Poisson distributed.

Poisson PMF





Poisson Distribution

- Limit of the Binomial distribution for low probability of success and large number of trials.
- That is, for rare events. 



The Poisson CDF

```
In [1]: samples = np.random.poisson(6, size=10000)
```

```
In [2]: x, y = ecdf(samples)
```

```
In [3]: _ = plt.plot(x, y, marker='.', linestyle='none')
```

```
In [4]: plt.margins(0.02)
```

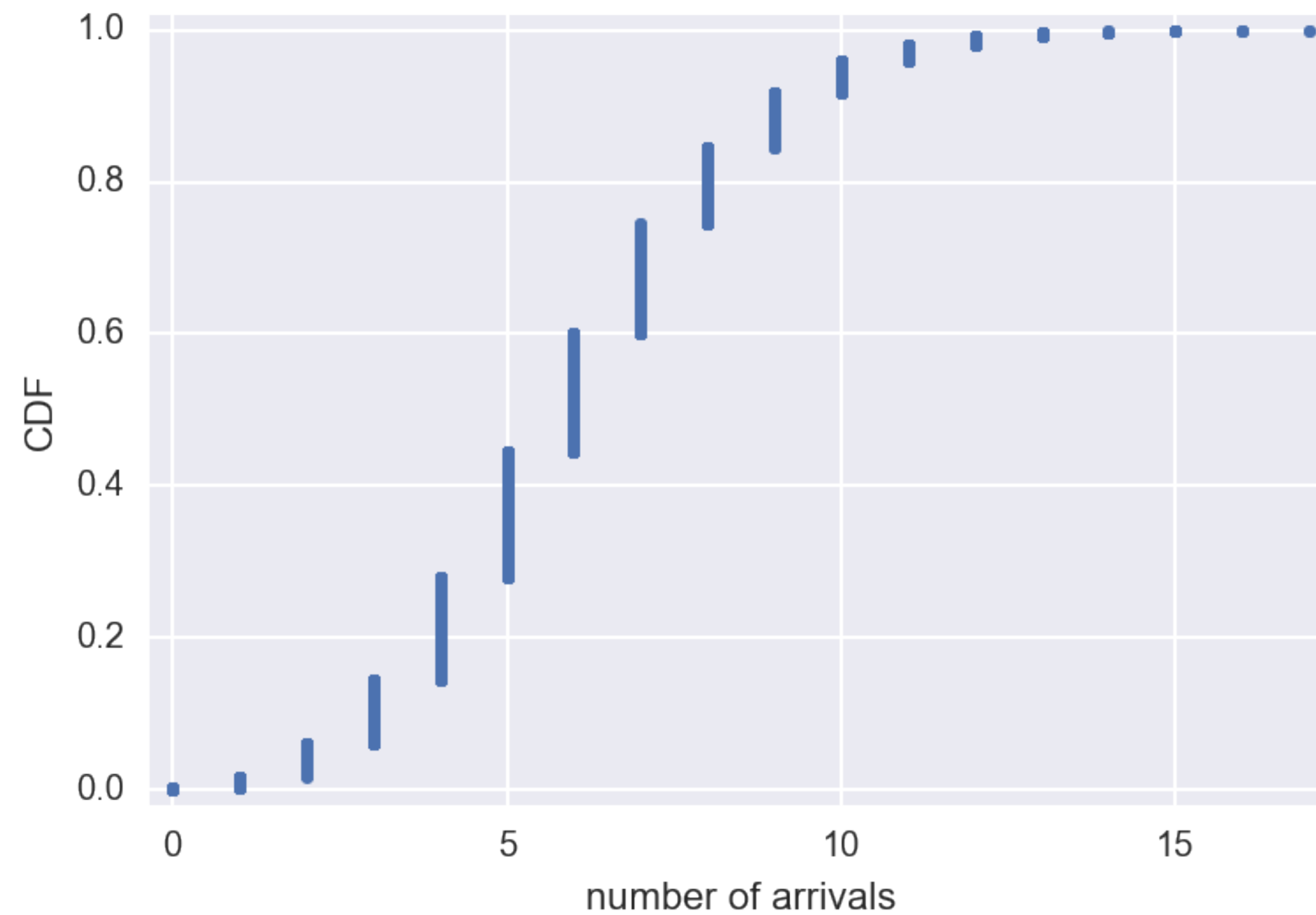
```
In [5]: _ = plt.xlabel('number of successes')
```

```
In [6]: _ = plt.ylabel('CDF')
```

```
In [7]: plt.show()
```



The Poisson CDF





STATISTICAL THINKING IN PYTHON I

Let's practice!