

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

def show_illustration(n_plots=20, extremes=False):
    std = 22 # Standard deviation of data
    xbar = 95 # Sample mean
    SE = std/np.sqrt(100) # Standard error
    print(SE)

    # What do confidence intervals mean?
    fig = plt.figure(figsize=np.array([4,2])*1.5, dpi=200)
    xq = np.linspace(xbar-3*SE, xbar+3*SE,100)
    ci = [xbar-2*SE, xbar+2*SE]
    rng = np.random.RandomState(0)
    if extremes:
        possible_hyp = ci # Extremes are at the end
    else:
        possible_hyp = rng.rand(n_plots)*(ci[1]-ci[0]) + ci[0]
    for i, mu_hypothesis in enumerate(possible_hyp):
        pdf_hypothesis = norm.pdf(xq, loc=mu_hypothesis, scale=SE)
        if i == 0:
            plt.plot(xq, pdf_hypothesis, 'k--', label='Possible true dist.')
        else:
            plt.plot(xq, pdf_hypothesis, 'k--')

    # Show xbar and confidence interval
    yl = [0, 0.25]
    plt.plot(xbar*np.ones(2), yl, 'r-', label='xbar')
    plt.plot(ci[0]*np.ones(2), yl, 'b-', label='95% Conf. Interval')
    plt.plot(ci[1]*np.ones(2), yl, 'b-')
    plt.ylim(yl)

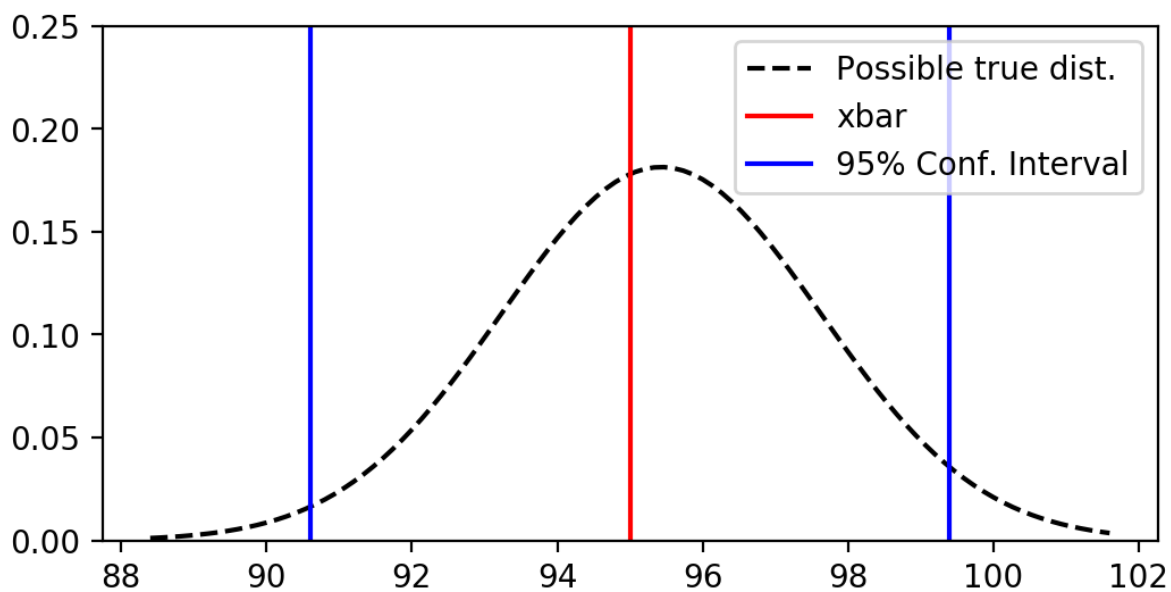
    plt.legend()
    plt.show()
show_illustration()
```

2.2

<Figure size 1200x600 with 1 Axes>

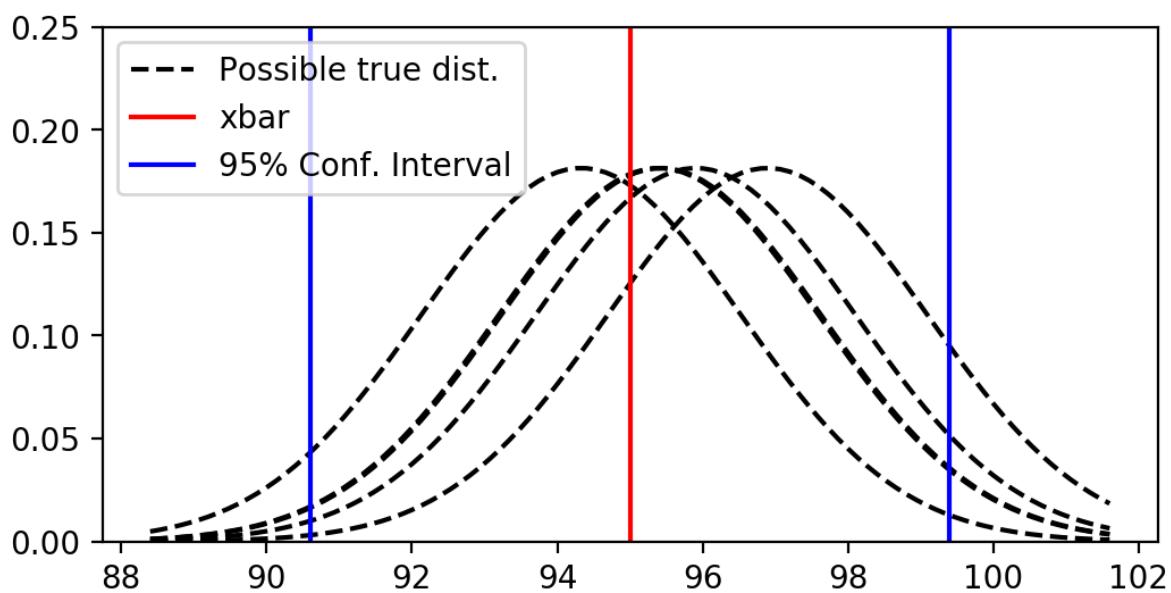
```
In [2]: show_illustration(1)
```

2.2



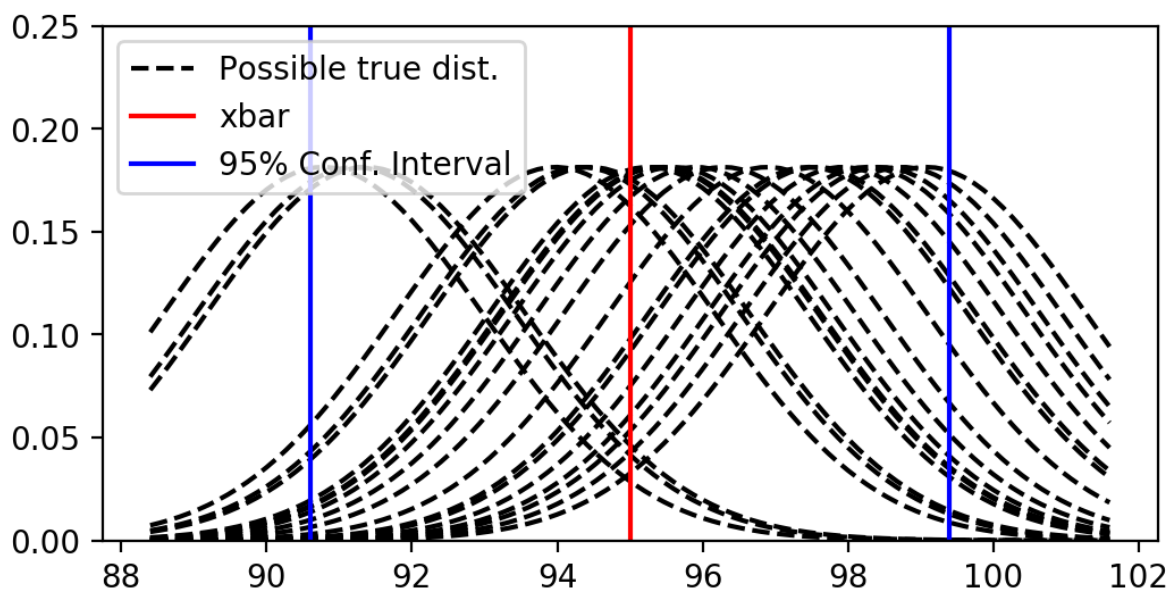
```
In [3]: show_illustration(5)
```

2.2



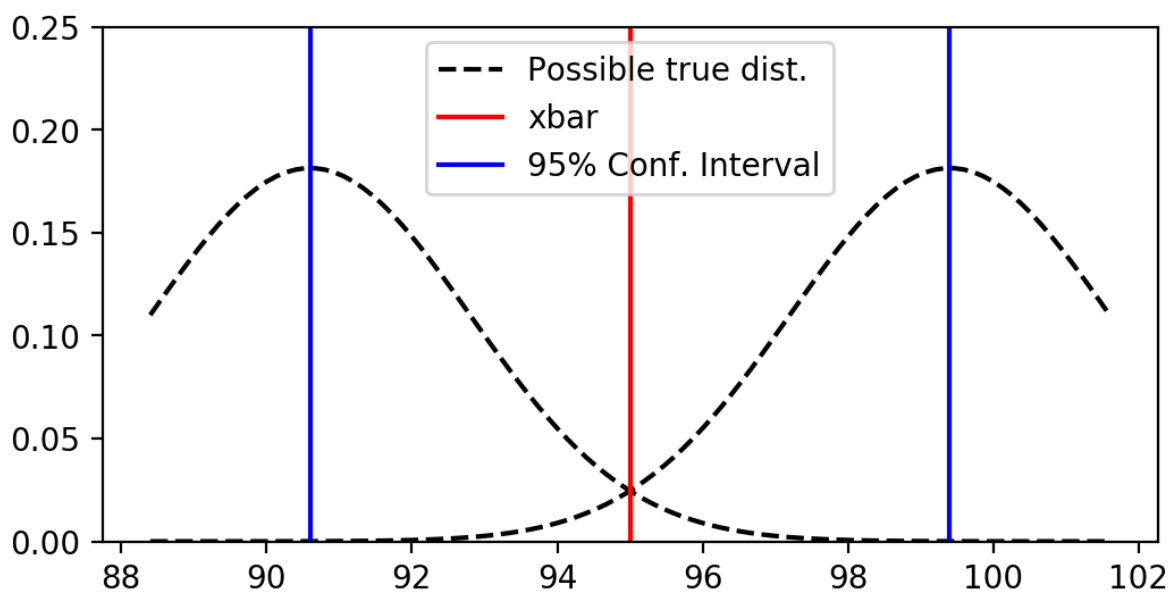
```
In [4]: show_illustration()
```

2.2



```
In [5]: show_illustration(extremes=True)
```

2.2



```
In [ ]:
```