Bayesian Analysis Explained Simply

1 What's the Problem?

Imagine you have 6 temperature measurements: 26.6, 38.5, 34.4, 34, 31, 23.6 degrees.

Model:

$$Y_1, \ldots, Y_6 \overset{\text{i.i.d}}{\sim} N(\theta, \sigma^2)$$

Translation: Each measurement follows a normal distribution with unknown average θ and unknown variability σ^2

You want to figure out:

- What's the **true average temperature** (called θ - "theta")?
- How much do the measurements **vary** around that average (called σ^2 - "sigma squared")?

Python Hands-on: Load the data and visualize it

```
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Load data
temps = np.array([26.6, 38.5, 34.4, 34, 31, 23.6])

# Step 2: Plot
plt.hist(temps, bins=5, color='skyblue', edgecolor='black')
plt.title('Histogram of Temperature Readings')
plt.xlabel('Temperature')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

Listing 1: Load and plot temperature data

2 The Bayesian Approach

Instead of just calculating averages like in regular statistics, Bayesian analysis asks: "What do we believe about these unknown values, and how does our belief change after seeing the data?"

3 Step 1: Starting Beliefs (Priors)

Before seeing any data, we say: "We have no idea what θ or σ could be - they could be almost anything within reason."

Prior:

$$\theta, \log \sigma \overset{\text{i.i.d}}{\sim} \text{Unif}(-C, C)$$

Translation: Both the average and the log of the standard deviation could be any value between $-C$ and $+C$ with equal probability (where C is very large)

Prior density:

$$f_{\theta,\sigma}(\theta,\sigma) = \frac{I\{-C < \theta, \log\sigma < C\}}{4\sigma C^2}$$

Translation: This is the mathematical way to write "we're equally uncertain about all reasonable values of θ and σ"

Python Tip: Simulate prior samples

```
C = 100
theta_prior = np.random.uniform(-C, C, 1000)
log_sigma_prior = np.random.uniform(-C, C, 1000)
sigma_prior = np.exp(log_sigma_prior)

# Visualize some priors
plt.hist(theta_prior, bins=30, alpha=0.7, label='theta')
plt.hist(sigma_prior, bins=30, alpha=0.7, label='sigma')
plt.title("Simulated Prior Distributions")
plt.legend()
plt.show()
```

Listing 2: Simulate prior samples

4 Step 2: What the Data Tells Us (Likelihood)

The data follows a **normal distribution** (bell curve). This means:

- Most measurements cluster around the true average
- Some measurements are a bit higher or lower
- Very few measurements are extremely far from the average

Likelihood:

$$f_{Y_1,\ldots,Y_n|\theta,\sigma}(y_1,\ldots,y_n) = (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\theta)^2\right)$$

Translation: This formula calculates how likely it is to observe our specific data values given particular values of θ and σ.

Python Hands-on: Compute log-likelihood

```
from scipy.stats import norm

def log_likelihood(theta, sigma, data):
    return np.sum(norm.logpdf(data, loc=theta, scale=sigma))

# Try different guesses
print(log_likelihood(theta=30, sigma=5, data=temps))
print(log_likelihood(theta=35, sigma=5, data=temps))
```

Listing 3: Compute log-likelihood of sample

5 Step 3: Combining Beliefs with Data (Posterior)

Bayesian analysis combines your starting belief with what the data shows to give you an **updated belief**.

Joint Posterior:

$$f_{\theta,\sigma|\text{data}}(\theta,\sigma) \propto I\{-C < \theta, \log\sigma < C\} \cdot \sigma^{-n-1} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\theta)^2\right)$$

Translation: This combines the prior and likelihood.

Python Hands-on: Visualize posterior landscape (grid approach)

```
theta_grid = np.linspace(25, 38, 100)
sigma_grid = np.linspace(1, 10, 100)
posterior = np.zeros((100, 100))

for i, theta in enumerate(theta_grid):
    for j, sigma in enumerate(sigma_grid):
        prior = 1 / sigma  # Improper uniform prior on log(sigma)
        like = np.exp(log_likelihood(theta, sigma, temps))
        posterior[i, j] = prior * like

# Normalize
posterior /= np.sum(posterior)

# Plot posterior
plt.imshow(posterior.T, extent=[25, 38, 1, 10], origin='lower', aspect='
    auto')
plt.colorbar(label='Posterior density')
plt.xlabel('Theta')
plt.ylabel('Sigma')
plt.title('Posterior over Theta and Sigma')
plt.show()
```

Listing 4: Posterior heatmap (grid approach)

6 Key Results

6.1 For the Average Temperature (θ):

To find just the posterior for θ, we integrate out σ:

$$f_{\theta|\text{data}}(\theta) \propto \left[\sum_{i=1}^{n}(y_i-\theta)^2\right]^{-n/2}$$

This can be rewritten as:

$$f_{\theta|\text{data}}(\theta) \propto \left(1+\frac{n(\bar{y}-\theta)^2}{(n-1)s^2}\right)^{-n/2}$$

Key Discovery:

$$\frac{\sqrt{n}(\theta-\bar{y})}{s}\bigg|\text{data} \sim t_{n-1}$$

Calculations with our data:

$$\bar{y} = \frac{26.6+38.5+34.4+34+31+23.6}{6} = 31.35$$

$$s = 5.48$$

Python Hands-on: Compute posterior probabilities using t-distribution

```
from scipy.stats import t

n = len(temps)
mean_y = np.mean(temps)
std_y = np.std(temps, ddof=1)

# 51% chance between 28 and 32
t_low = (28 - mean_y) * np.sqrt(n) / std_y
t_high = (32 - mean_y) * np.sqrt(n) / std_y

p_between = t.cdf(t_high, df=n-1) - t.cdf(t_low, df=n-1)
print(f"Probability theta is between 28 and 32: {p_between:.3f}")
```

Listing 5: t-distribution posterior for theta

6.2 For the Variability (σ):

To find the posterior for σ, we integrate out θ:

$$f_{\sigma|\text{data}}(\sigma) \propto I\{\sigma > 0\}\sigma^{-n} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right)$$

Let

$$w = \frac{(n-1)s^2}{\sigma^2}$$

Then:

$$w \sim \chi^2_{n-1}$$

Python Hands-on: Plot posterior for σ

```
from scipy.stats import chi2

w_vals = chi2.rvs(df=n-1, size=10000)
sigma_samples = np.sqrt((n-1)*std_y**2 / w_vals)

plt.hist(sigma_samples, bins=30, density=True, color='salmon')
plt.title("Posterior Distribution of Sigma")
plt.xlabel("Sigma")
plt.ylabel("Density")
plt.show()
```

Listing 6: Posterior density for sigma

Point estimate:

$$\hat{\sigma} = s = 5.48$$

7 Why This Matters

Regular statistics just gives you point estimates. Bayesian analysis tells you:

1. Your best guess for the unknown values
2. How uncertain you should be about those guesses
3. The probability that the true values fall in any range you care about

8 Real-World Translation

If these were daily temperature readings:

- "The average daily temperature is probably around 31.4 degrees"
- "But I'm 95% confident it's somewhere between 25.6 and 37.1 degrees"
- "There's about a 50-50 chance it's in the comfortable 28-32 degree range"

This gives you a much richer understanding than just saying "the average is 31.4 degrees" and stopping there.