UNIVERSITY OF MARYLAND

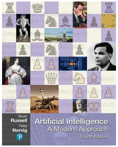
MSML610: Advanced Machine Learning

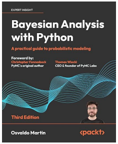
7.2: Posterior-based Decisions

Instructor: Dr. GP Saggese - gsaggese@umd.edu

References:

- AIMA (Artificial Intelligence: a Modern Approach)
  - Chap 15: Probabilistic programming
- Martin, Bayesian Analysis with Python, 2018 (2e)





SCIENCE ACADEMY

- ***Posterior-Based Decisions***
  - Chemical Shift: Example
  - Posterior Predictive Checks
  - Groups Comparison

Posterior-Based Decisions

- Sometimes describing the posterior is not enough
  - **Make decisions based on inference**
- E.g., is the coin fair ($\theta = 0.5$) or biased?
  - $\mathbb{E}[\hat{\theta}] = 0.324$ suggests bias
  - Can't rule out unbiased since:
    - $HPI = [0.03, 0.65]$
    - $0.5 \in HPI$
  - For **sharper decisions**:
    - Collect more data to reduce posterior spread
    - Define a more informative prior

Savage-Dickey Density Ratio

- **Savage-Dickey ratio** tests *point null-hypotheses* in Bayesian inference
- **Idea**: compare prior and posterior densities at a single point θ_0

$$BF_{01} = \frac{p(\theta_0|H_1)}{p(\theta_0|\mathcal{D}, H_1)}$$

where:

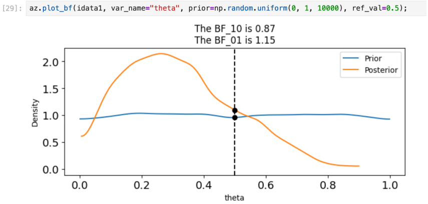
- $p(\theta_0|H_1)$ is *prior* density θ under alternative hypothesis H_1, evaluated at θ_0
- $p(\theta_0|\mathcal{D}, H_1)$ is *posterior* density θ under H_1 evaluated at θ_0

- **Intuition**: show how data changes belief about θ_0
  - If posterior density at θ_0 is much smaller than prior density, strong evidence against null hypothesis H_0
- **Cons**:
  - It's a point statistics (not considering the entire posterior)

Bayes Factor (BF)	**Interpretation**
1 - 3	Not enough evidence
3 - 10	Substantial evidence
10 - 100	Strong evidence
$>$ 100	Decisive evidence

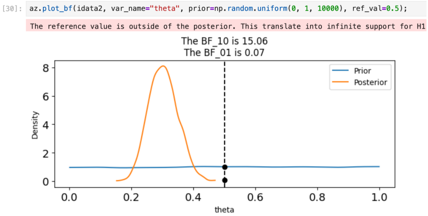
SCIENCE ACADEMY

Savage-Dickey Density Ratio: Example



```
[29]: az.plot_bf(idata1, var_name="theta", prior=np.random.uniform(0, 1, 10000), ref_val=0.5);
```

The BF_10 is 0.87
The BF_01 is 1.15



```
[30]: az.plot_bf(idata2, var_name="theta", prior=np.random.uniform(0, 1, 10000), ref_val=0.5);
```

```
The reference value is outside of the posterior. This translate into infinite support for H1
```

The BF_10 is 15.06
The BF_01 is 0.07

- H_0: “coin is fair”
  - The prior is 0.87
  - The posterior is 1.15
  - $BF_{10} = 0.76 \rightarrow$ **no evidence** to reject the null hypothesis
- In the other case
  - $BF_{10} = 15.06 \rightarrow$ **strong evidence** to reject the null hypothesis

Region of Practical Equivalence

- **Region of Practical Equivalence** (ROPE) $=$ interval where values are "equivalent"
- **Example**
  - H_0: *"coin is fair"* impractical if $\theta = 0.5$
  - ROPE: $\theta \in [0.45, 0.55]$ equivalent to 0.5
- **Hypothesis testing with ROPE and HPI**
  - Compare ROPE with HPI (Highest-Posterior Interval)
    - HPI within ROPE → no effect, reject H_1
    - HPI outside ROPE → effect present, reject H_0
    - HPI overlaps ROPE → inconclusive
- **Decide ROPE before analysis based on domain knowledge**
  - Picking it after analysis is like choosing p-value threshold after seeing p-value

Loss Function: Motivation

- For many problems:
  - **Make decisions based on inference**
  - **Decision cost is asymmetric**
    - Cost of a bad decision $>$ or $<$ benefit of a good decision
    - Vaccines may cause overreaction, but benefits outweigh risks
- Measure for **best decision**:
  - Benefits of a correct decision
  - Cost of a mistake
  - Trade-off between benefits and costs using a loss function
  - Use loss function for decisions

Loss Function

- Aka “cost function”
  - The inverse is known as “objective”, “fitness”, “utility function”
- **Loss function** quantifies *“how bad is an estimation mistake?”*
  - Larger loss indicates worse estimation
  - Loss is difference between:
    - The true value θ; and
    - The estimated value $\hat{\theta}$

Loss	**Expression**	**Point estimate**
Quadratic loss	$(\theta - \hat{\theta})^2$	Mean of posterior
Absolute loss	$\lvert\theta - \hat{\theta}\rvert$	Median of posterior
1-0 loss	$I(\theta \neq \hat{\theta})$	Mode of posterior

- Algorithm to make decisions in Bayesian statistics using loss function
  - Goal: pick a single value $\hat{\theta}$
  - True value θ is unknown
  - Estimate θ using posterior distribution
  - Find $\hat{\theta}$ minimizing expected loss function

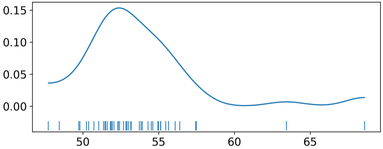
Tutorial

- Bayesian Coin

- Posterior-Based Decisions
    - ***Chemical Shift: Example***
    - Posterior Predictive Checks
    - Groups Comparison

Chemical Shift

- **Nuclear magnetic resonance** (NMR)
  - Used to study molecules of living things
  - Measures observable quantities related to unobservable molecular properties (e.g., chemical shift)
- **Chemical shift** reveals local magnetic environment of atomic nuclei
  - Identifies molecular structure and functional groups
- **Example of chemical shift**
  - Data looks Gaussian with a couple of outliers



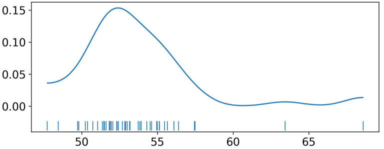
Use of Gaussians in Statistics

- Aka “normal” distribution
- **Gaussians** are:
  - easy to work with
  - abundant in nature
- **Pros**:
  - Average of large sample size tends to be Gaussian (Central Limit Theorem)
  - Many phenomena approximated using Gaussians (since they are average of effects)
  - Conjugate prior of Gaussian is Gaussian
- **Cons**:
  - Not robust to outliers
  - Important to relax assumption of Gaussianity

Chemical Shift: Example

- **Assume Gaussian** approximates data from example chemical shift
- **Likelihood** from normal distribution:

$$Y \sim N(\mu, \sigma)$$



- Weakly informative **priors** for mean and sigma of Y
  - **Mean** from uniform distribution

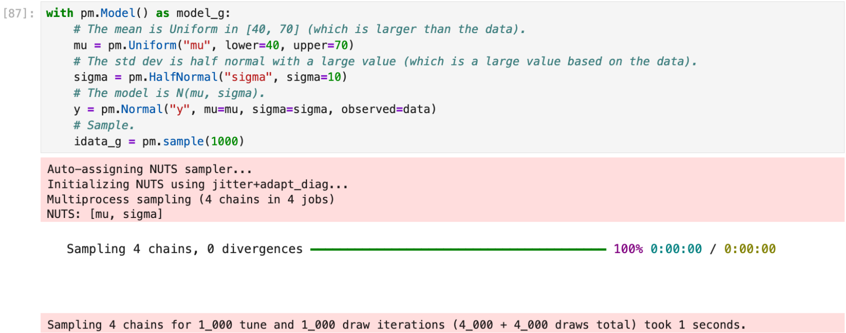
$$\mu \sim U(l = 40, h = 70)$$

    - Larger than data range, e.g., $[40, 70]$
  - **Std dev** from half-normal distribution

$$\sigma \sim HalfNormal(0, \sigma_\sigma = 10)$$

SCIENCE ACADEMY

Chemical Shift: PyMC



```
[87]: with pm.Model() as model_g:
          # The mean is Uniform in [40, 70] (which is larger than the data).
          mu = pm.Uniform("mu", lower=40, upper=70)
          # The std dev is half normal with a large value (which is a large value based on the data).
          sigma = pm.HalfNormal("sigma", sigma=10)
          # The model is N(mu, sigma).
          y = pm.Normal("y", mu=mu, sigma=sigma, observed=data)
          # Sample.
          idata_g = pm.sample(1000)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [mu, sigma]
```

```
 Sampling 4 chains, 0 divergences ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 / 0:00:00
```

```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_000 draws total) took 1 seconds.
```

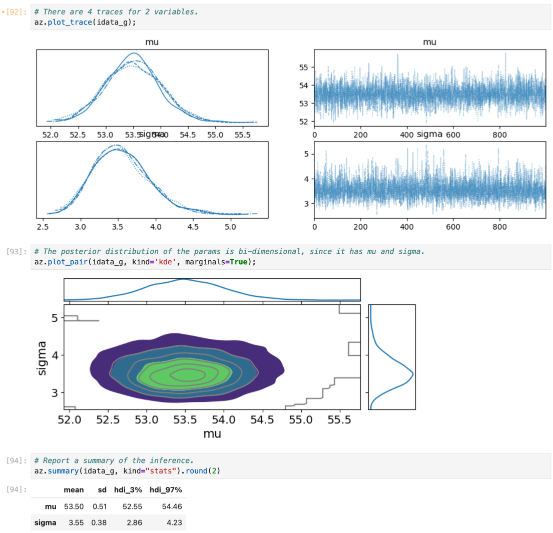
- The PyMC code is **one-to-one with the model**:

$$\begin{cases} \mu \sim U(l = 40, h = 70) \\ \sigma \sim HalfNormal(0, \sigma_\sigma = 10) \\ Y \sim N(\mu, \sigma) \end{cases}$$

  - Get 1000 samples from the posterior

Chemical Shift: PyMC

- Compute 4 traces for 2 variables μ, σ
  - Results are well-formed
  - $\mu \in [52.55, 54.45]$
  - $\sigma \in [2.86, 4.23]$
- **Is the model good?**



```
•[92]: # There are 4 traces for 2 variables.
       az.plot_trace(idata_g);
```

```
[93]: # The posterior distribution of the params is bi-dimensional, since it has mu and sigma.
      az.plot_pair(idata_g, kind='kde', marginals=True);
```

```
[94]: # Report a summary of the inference.
      az.summary(idata_g, kind="stats").round(2)
```

[94]:

	mean	sd	hdi_3%	hdi_97%
mu	53.50	0.51	52.55	54.46
sigma	3.55	0.38	2.86	4.23

- Posterior-Based Decisions
  - Chemical Shift: Example
  - ***Posterior Predictive Checks***
  - Groups Comparison

Samples from Posterior Distribution

- Given **posterior distribution** $\Pr(\theta|y)$, generate predictions $\tilde{y}$ based on data y and estimated parameters $\hat{\theta}$:

$$\Pr(\tilde{y}|\mathrm{y}) = \int_{\hat{\theta}} \Pr(\tilde{y}|\theta)\Pr(\theta|y)d\theta = \int \text{model} \times \text{posterior}$$

  - This is called the *"posterior predictive distribution"* as it predicts *future data* using the *posterior distribution*
- **Conceptually**
  - Sample a value of θ from the posterior $\Pr(\theta|y)$
  - Feed the value of θ to the likelihood $\Pr(\tilde{y}|\theta)$
  - Obtain $\tilde{y}$
- This process has **two sources of uncertainty**:
  - Parameter uncertainty
    - Captured by the posterior $\Pr(\theta|y)$
  - Sampling uncertainty
    - Captured by the likelihood $\Pr(\tilde{y}|\theta)$

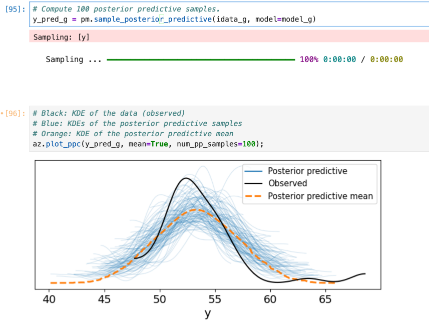
Posterior Predictive Check (PPC)

- **PPC approach**:
  - Generate predictions $\tilde{y}$ with observed data y from posterior distribution
  - Check consistency between predictions and observed data
- **Intuition**: can the model reproduce observed data?
  - Always check!
- **Differences** arise from:
  - Mistakes
  - Model limitations
    - E.g., works for average behavior but fails for rare values
  - Data limitations

Bayesian Workflow Using PPC

1. Given a process, **true distribution** is unknown / unknowable
2. **Sample the process**
   - Get finite sample y
   - E.g., experiment, survey, simulation
3. **Inference**
   - Build probabilistic model using prior $\Pr(\theta)$ and likelihood $\Pr(y|\theta)$ to get posterior $\Pr(\theta|y)$
   - Posterior is distribution of model parameters θ given data
4. **Predictive distribution**
   - Compute predictions from posterior (posterior predictive distribution)
   - Posterior predictive is distribution of predicted samples averaged over posterior
5. **Validation**
   - Validate model by comparing original vs predicted samples

Chemical Shift Example: PPC



```
[95]: # Compute 100 posterior predictive samples.
      y_pred_g = pm.sample_posterior_predictive(idata_g, model=model_g)
```

```
Sampling: [y]

    Sampling ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 / 0:00:00
```

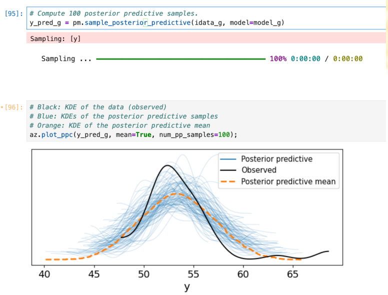
```
[96]: # Black: KDE of the data (observed)
      # Blue: KDEs of the posterior predictive samples
      # Orange: KDE of the posterior predictive mean
      az.plot_ppc(y_pred_g, mean=True, num_pp_samples=100);
```

- **Apply the Bayesian workflow**
  - Sample posterior
  - Apply model
  - Get predictive posterior distribution (dashed orange)
  - Compare to data (black)
- **Is the PPC model good?**
  - **No!**
  - Posterior mean is right of data
  - Posterior std dev is larger
  - You can't generate the data from the model!

SCIENCE ACADEMY

Chemical Shift: Model Critique

- **Problem**
  - Two data points on distribution tails
  - Normal distribution:
    - is “surprised” by these points
    - “reacts” by adjusting mean towards them and increasing the standard deviation



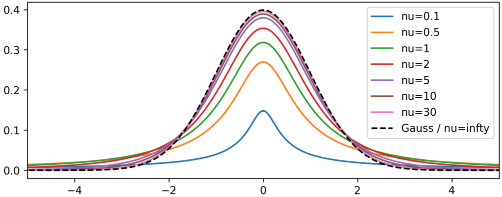
```
[95]: # Compute 100 posterior predictive samples.
      y_pred_g = pm.sample_posterior_predictive(idata_g, model=model_g)

      Sampling: [y]

        Sampling ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 / 0:00:00
```

```
•[96]: # Black: KDE of the data (observed)
       # Blue: KDEs of the posterior predictive samples
       # Orange: KDE of the posterior predictive mean
       az.plot_ppc(y_pred_g, mean=True, num_pp_samples=100);
```

- **Possible solutions**
  1. Declare and discard outliers
     - E.g., equipment malfunction (requires evidence)
  2. Change the model
  3. **Bayesian philosophy**
     - Encode assumptions into the model (e.g., priors, likelihoods)
     - Avoid ad-hoc heuristics (e.g., outlier removal rules)

Student’s t-distribution: Recap

- **Student’s t-distribution** has 3 params:
  1. Mean μ
     - Doesn’t always exist
  2. Scale σ
     - Similar to std dev, but doesn’t always exist
  3. Degrees of freedom $\nu \in [0, \infty]$
     - “Normality parameter”: controls distribution normality
     - $\nu = 1$: heavy tails, no mean (Cauchy)
     - $\nu \to \infty$: Gaussian
- Student’s t has **heavy tails** (high kurtosis)
  - Values more likely far from mean compared to Normal

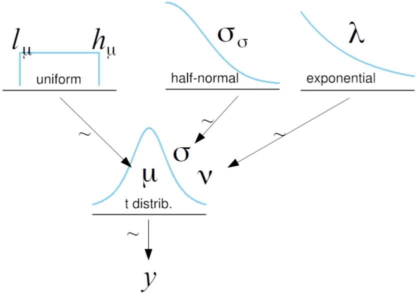


SCIENCE
ACADEMY

Chemical Shift: Use Student's t-dist (1/3)

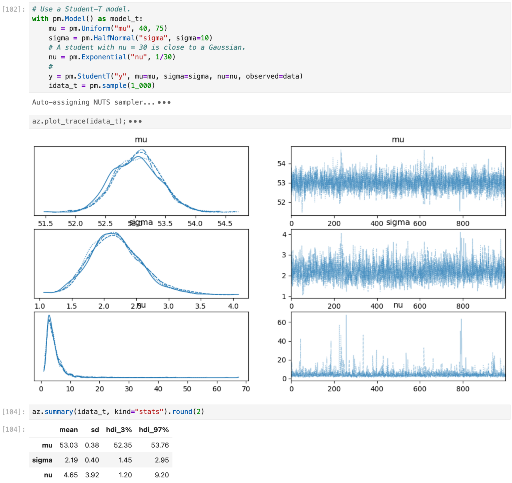
- Use Student's t-distribution instead of Normal in model for chemical shift

$$\begin{cases} \mu \sim U(l, h) \\ \sigma \sim HalfNornal(0, \sigma) \\ \nu \sim Exp(\lambda) \\ y \sim StudentT(\mu, \sigma, \nu) \end{cases}$$



Chemical Shift: Use Student's t-dist (2/3)

- Decrease ν (**less Gaussian**) instead of increasing mean and standard deviation
    - μ similar to Gaussian estimate
    - σ smaller
    - $\nu \approx 5$ (not very Gaussian)
- **Estimation more robust**
    - Outliers have less effect



```
[102]: # Use a Student-T model.
       with pm.Model() as model_t:
           mu = pm.Uniform("mu", 40, 75)
           sigma = pm.HalfNormal("sigma", sigma=10)
           # A student with nu = 30 is close to a Gaussian.
           nu = pm.Exponential("nu", 1/30)
           #
           y = pm.StudentT("y", mu=mu, sigma=sigma, nu=nu, observed=data)
           idata_t = pm.sample(1_000)

       Auto-assigning NUTS sampler...•••

       az.plot_trace(idata_t);•••
```

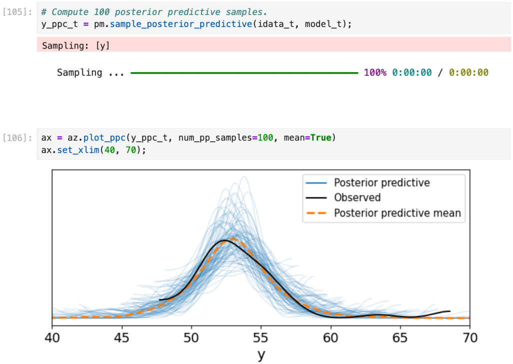
```
[104]: az.summary(idata_t, kind="stats").round(2)
```

[104]:

	mean	sd	hdi_3%	hdi_97%
mu	53.03	0.38	52.35	53.76
sigma	2.19	0.40	1.45	2.95
nu	4.65	3.92	1.20	9.20

SCIENCE ACADEMY

Chemical Shift: Use Student’s t-dist (3/3)



```
[105]: # Compute 100 posterior predictive samples.
       y_ppc_t = pm.sample_posterior_predictive(idata_t, model_t);
```

```
Sampling: [y]

Sampling ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 / 0:00:00
```

```
[106]: ax = az.plot_ppc(y_ppc_t, num_pp_samples=100, mean=True)
       ax.set_xlim(40, 70);
```

- PPC fits better than Normal model
- Plot is “hairy” because KDE is estimated only in data interval and 0 outside

Tutorial

- Robust modeling

- Posterior-Based Decisions
    - Chemical Shift: Example
    - Posterior Predictive Checks
    - ***Groups Comparison***

Group Comparison

- **Group comparison** tests for statistically significant results between *treatment* and *control group*
- E.g.,
  - *How well do patients respond to a new drug vs a placebo?*
  - *Is there a reduction in car accidents after new traffic regulation?*
  - *Does college student performance improve without cellphones at school?*
- **Effect size** quantifies the difference between two groups
  - From *"does it work?"* (hypothesis testing) to *"how well does it work?"* (estimate effect size)

Bogus Control Groups

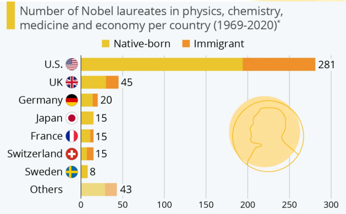
- **Always ask for the baseline** used for comparison, when something is claimed to be harder/better/faster/stronger
  - E.g., sell sugary yogurts to boost the immune system by comparing it to milk
    - Better control group: less sugary yogurt
- **Placebo** is a psychological phenomenon where a patient experiences improvements after receiving an inactive treatment
  - Using a placebo is better than “no treatment”
  - Shows difficulty in accounting for all factors in an experiment

Group Comparison Bayesian-Style

- **Frequentist approach**
  - Compare p-value of difference of means in each group
- **Bayesian approach**
  - Compare posterior distribution of means between groups using:
    - Plot of posterior
    - Cohen's d
    - Probability of superiority

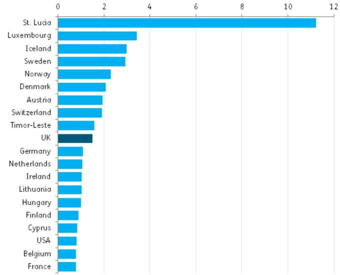
Sample Size Effect

- **Sample size effect** is the impact of the number of observations on statistical results (e.g., p-values, confidence intervals)
- **Small sample effect**
  - Large mean difference might not be statistically significant (low p-value)
  - Estimates (e.g., means, proportions, correlations) fluctuate widely due to high sampling variability
  - Outliers have a disproportionate influence
  - Inference is unstable: results may not replicate
- **Large sample**:
  - Tiny mean difference can be highly significant (small p-value) but meaningless
- E.g., Cohen's d, probability of superiority



Number of Nobel laureates in physics, chemistry, medicine and economy per country (1969-2020)*

Nobel Laureates per million 2016 population by country of birth (top 20 countries)

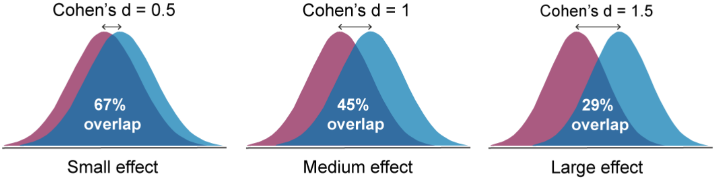


Cohen’s d

- **Cohen’s d** is the difference of means relative to pooled standard deviation

$$\frac{\mu_2 - \mu_1}{\sqrt{(\sigma_1^2 + \sigma_2^2)/2}}$$

  - Normalizes effect by variability for pooled std dev
  - Variability of each group normalizes mean difference
  - Similar to a Z-score, number of std dev values differ



- **Bayesian approach**
  - Compute posterior distribution of means and std → formula
  - Compute distribution of Cohen’s d → summary statistics