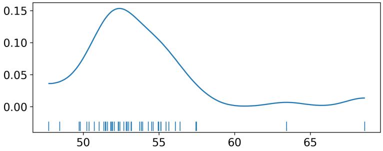
Tutorial: Bayesian Coin

- Bayesian Coin

- ***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)
- Data looks Gaussian with a couple of outliers



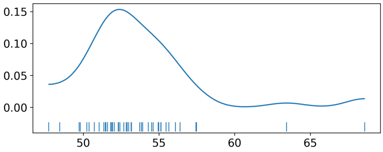
Use of Gaussians in Statistics

- Aka “normal” distribution
- **Gaussians are easy to work with and abundant in nature**
- **Pros**:
  - Average of large sample size tends to be Gaussian (by 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 is a decent approximation of the data
- The **likelihood** $y|\mu, \sigma$ comes from a normal distribution:

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



- **Priors** for mean and sigma of Y
  - **Mean** from uniform distribution $U(l, h)$:

$$\mu \sim U(l, h)$$

    - Set μ larger than data range, e.g., $[40, 70]$
  - Std dev from half-normal distribution (i.e., a regular normal, but restricted to non-negative values):

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

    - If unknown, set $\sigma_\sigma = 10$

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**:

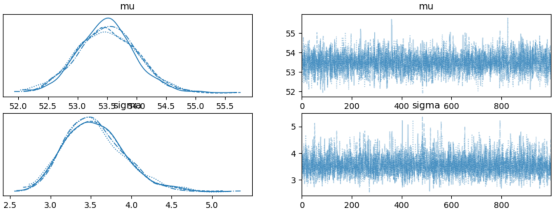
$$\begin{cases} \mu \sim U(l = 40, h = 70) \\ \sigma \sim HalfNormal(0, \sigma_\sigma = 10) \\ Y \sim N(\mu, \sigma) \end{cases}$$

  - Get 1000 samples from the posterior

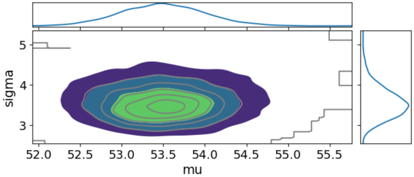
Chemical Shift: PyMC

- 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

SCIENCE ACADEMY

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

Samples from Posterior Distribution

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

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

- 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 $\Pr(y|\theta)$
  - 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 $\Pr(y|\theta)$

SCIENCE ACADEMY

Posterior Predictive Check (PPC)

- **Intuition**: can the model reproduce observed data?
- **PPC approach**:
  - Generate predictions $\tilde{y}$ with observed data y
  - Check consistency between predicted values and observed data
- Models should always be checked
- Differences can arise by:
  - Mistakes
  - Limitations of the model
    - E.g., the model works well for average behavior but fails to predict rare values
  - Limitations of the data

Bayesian Workflow Using PPC

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

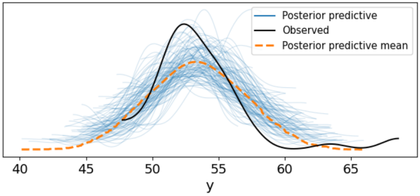
Chemical Shift Example: PPC

[95]:
```python
# 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]:
```python
# 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);
```



- Sample the posterior
- Apply the model
- Get predictive posterior distribution (dashed orange distribution)
- Compare to data (black distribution)
- **Is the PPC model good?**
- **No**
  - Posterior mean is more to the right than data
  - Posterior std dev is larger

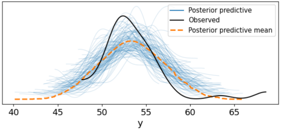
Chemical Shift: Model Critique

[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);
```

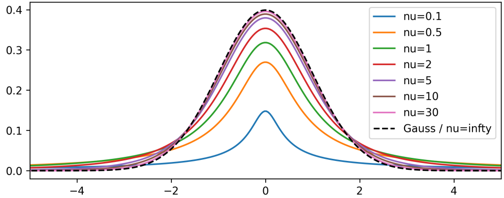


- **Problem**
  - Two data points on the tails of the distribution
  - The normal distribution is "surprised" by these points and "reacts" by adjusting the mean towards them and increasing the standard deviation

- **Solution**
  - Declare points as outliers and discard
    - E.g., equipment malfunction (need evidence)
  - Change the model
- Bayesians prefer to encode assumptions into the model (e.g., priors, likelihoods)
  - Rather than ad-hoc heuristics (e.g., outlier removal rules)

Student's t-distribution: Recap

- Student's t-distribution has 3 params:
  1. Mean μ
     - It doesn't always exist
  2. Scale σ
     - Similar to std dev, but it doesn't always exist
  3. Degrees of freedom $\nu \in [0, \infty]$
     - Aka "normality parameter", since it controls how "normal" is the distribution
     - With $\nu = 1$: heavy tails and no mean (Cauchy)
     - With $\nu \to \infty$ we recover the Gaussian
- Heavy tails (high kurtosis) means *"values are more likely to be far from the mean compared to a Normal"*

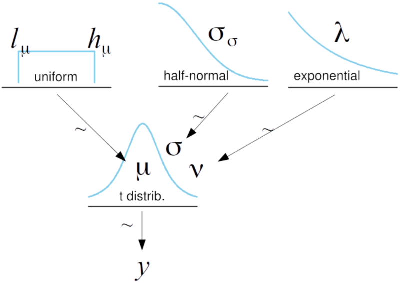


SCIENCE ACADEMY

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

- Use Student's t-distribution model instead of Normal

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



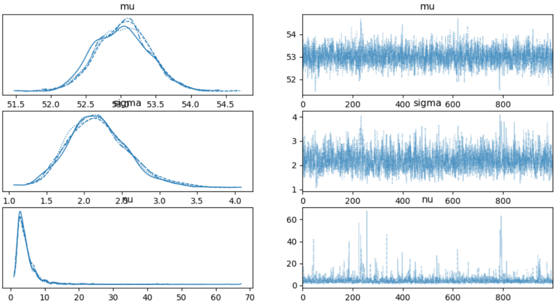
SCIENCE ACADEMY

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

[102]:
```python
# 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... •••
```

```python
az.plot_trace(idata_t); •••
```



[104]:
```python
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

- Outliers 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

UNIVERSITY OF MARYLAND 1856 SCIENCE ACADEMY

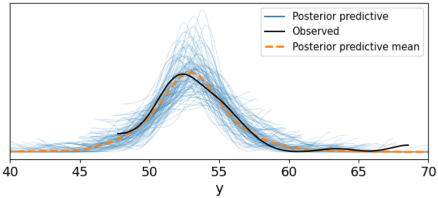
Chemical Shift: Use Student's t-dist (3/3)

[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 (posterior predictive check) fits better than Normal model
- Plot is "hairy" because KDE is estimated only in data interval and 0 outside

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

SCIENCE ACADEMY

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
  - Moves from "does it work?" (hypothesis testing) to "how well does it work?" (estimate effect size)

Bogus Control Groups

- When something is claimed to be harder/better/faster/stronger, ask for the baseline used for comparison
  - E.g.,
    - Sell sugary yogurts to boost the immune system by comparing it to using milk
    - A better control group would be a 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: Large mean difference might not be statistically significant (low p-value)
  - Large sample: Tiny mean difference can be highly significant (small p-value) but meaningless
  - E.g., Cohen's d

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
  - Bayesian analysis estimates actual std dev
- Variability of each group normalizes mean difference
  - Similar to a Z-score, number of std dev values differ
- Compute posterior distribution of means and std
- Compute distribution of Cohen's d or use formula