Introduction

We will discuss what I learned from the four tutorials: Bayesian Coin, Robust Modeling, Hierarchical Models, and Generalized Linear Models. I attempted to work through each of the notebooks to understand how they flow.

1. Bayesian Coin

We begin with the very simple but very useful example of flipping a coin to estimate how biased it is. The point is to assume that p, the probability of seeing heads, is itself a random variable, and to try to change our beliefs about p based on actual flips.
This assignment involves modeling using Bernoulli likelihoods and beta priors for each toss. The great thing about this is that beta is the conjugate prior for Bernoulli, which means that beta is also the form of my posterior distribution. Yeah, I actually rather liked seeing how Bayes updating is so straightforward algebraically.
Key concepts I learned:

1. Prior distribution signifies our beliefs prior to observations.
2. The probability from flips to updates the belief

One can easily conceptualize how uncertainty can decrease with added data.

2. Robust Modeling

Unlike in the coin example, in this case we are working with continuous data. The important thing to know about this is that sometimes data has outliers, and the normal likelihood function isn't sufficient because outliers will draw the mean too strongly.
Therefore, to correct this issue, Student-t distribution is employed by the tutorial since its tails are heavier. I learned that:

1. Gaussians tend to assume light tails to support observations
2. The t-Distribution deals with outliers way better
3. Degrees of freedom parameter(v) affects how heavy the tails are

The approach to modeling in PyMC (or other lib) has remained essentially the same: priors, likelihood, sample from posterior.
One thing I observed: the posterior estimates for μ and σ are quite stable even for a few problematic data points, which is kind of indicative of why robust modeling is needed for real data. It means that a few problematic data points won’t mess up everything.

3. Hierarchical Models

This question is the most interesting to me as it has introduced partial pooling. This is where we compare various groups, for example, classrooms or hospitals, which contain limited data.
Estimating each group completely independently would give very noisy estimates. Estimating entirely from the pooled data would ignore between-group differences. So, hierarchical modeling provides us with estimates lying in between:
Each group has its own parameter. However, these parameters are derived from a common distribution at a higher level.
It has a shrinkage effect on data, wherein the extreme groups are attracted towards the global mean.
This prevents overfitting and allows for more stable estimates, especially for groups with very few observations.
This also enabled me to appreciate how priors are important, how the group hyper priors control how much prior information is pooled. This also highlighted how often, in practical situations rather than in idealized examples, you find yourself dealing with hierarchical models.

4. GLMs

Lastly, the GLM tutorial provides examples on how to generalize regression. An important thing to remember about GLMs is that they can be used to handle various kinds of data by relating a linear predictor to the output using a link function.

Examples:
Logistic regression has a logit link
Poisson regression applies a log link function
Regular linear regression is equivalent to using the identity link function in GLMs.
The notebook illustrates a synthetic example, where we:

1. Define Predictors
2. Build a Linear Model
3. Choose a suitable likelihood
4. Fit it in Bayesian manner

One thing I learned is that GLMs combine everything from earlier lessons priors, likelihoods, inference, etc. but do so in non-Gaussian environments.
GLMs are super flexible and appear everywhere in applied modeling economics, biomedical stats, etc. The Bayesian approach provides estimates for uncertainty, which are not provided by classical GLMs.
Another thing that I observed was how notebooks focus on visualization after posterior plots, traceplots, etc., and it is very helpful to understand how the model behaves.
To be honest, I did find a few places where I had to read it twice for understanding, but once I ran the code for it, most things got clarified to me. I felt that these examples provide great

guidance on how to transition eventually from basic probability models to advanced Bayesian analyses.

5. Conclusion

The examples in the series also gave me a much better sense of how the Bayesian approach to modeling proceeds from toy problems to realistic modeling. The big ideas are uncertainty, updating beliefs, robustness, sharing strength between groups, generalizing regression seem well-integrated. However, I do want to say that I now understand, at least on an conceptual level, what each tutorial is trying to cover, even though certainly more could be learned about it on my part by going further into the mathematics or utilizing additional data.