Example of Bayesian reasoning

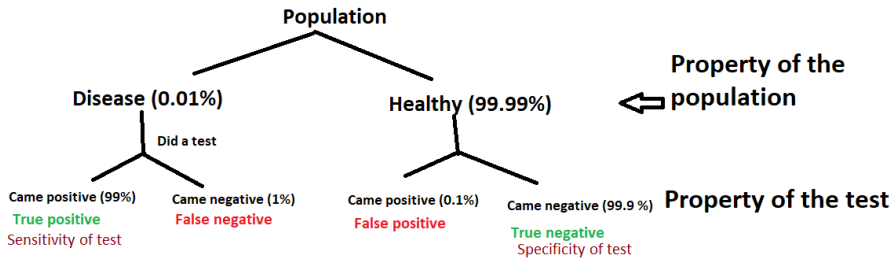

Figure 1: Testing for a "rare" disease scenario

Say someone came back positive. Commonly, that person will be considered among the affected individuals. Quantitatively speaking, we will in general set P(the event that the person is infected) = 1. But let's apply Bayes theorem and compute following probability.

Sensitivity= P(+ve|Disease) and Specificity= P(-ve|Healthy).
P(Got the disease | tested +ve) = $\frac{\text{P(+ve|Disease)P(Disease)}}{P(+ve)}$.

Now P(+ve) = P(+ve|Disease)P(Disease) + P(+ve|Healthy)P(Healthy) = (0.99*0.0001 + 0.001*0.9999)=0.0010989.

So, P(Got the disease | tested +ve) = 0.99*0.0001 / 0.0010989 ≈0.09.

This is an application of Bayes theorem, which is the building block for Bayesian inference.

The formula used to calculate P(+ve) is $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$ is Law of Total Probability. Here B^c is the complement set of B.

A general statistical analysis

- Assign a model for the data. It may be a supervised model, unsupervised model. (Or may be a semi-supervised model as well) **This part is in general independent of the inference plan adopted in the next stage.**
- Adopt an inference plan (parameter estimation and hypothesis testing)

Let θ be a parameter of interest. We usually come up with a function $\delta(\mathbf{y})$ of the data $\mathbf{y} = (y_1, \ldots, y_n)$ such that an estimate of θ is $\hat{\theta} = \delta(\mathbf{y})$.

Example: For a model $y_i = \theta + \epsilon_i$ with $\epsilon_i \sim \text{Normal}(0, \sigma^2)$ or equivalently $y_i \sim \text{Normal}(\theta, \sigma^2)$, a possible estimate of θ is $\hat{\theta} = \bar{\mathbf{y}} = \frac{1}{n}\sum_i y_i$ which is the $\delta(\mathbf{y})$. For different data $\mathbf{y}$, one will get

different estimates of θ.

The next step is to derive a distribution for $\delta(\mathbf{y})$.

Frequentist strategy: Note that the data $\mathbf{y}$ are coming from a random process. So, it will automatically induce a distribution for $\delta(\mathbf{y})$. Oftentimes, they do not require any explicit probability model for $\mathbf{y}$. Then, the strategy is to apply different versions of the Central Limit Theorem.

So, what are we looking for in order to make an inference? $\rightarrow$ Get a distribution for $\hat{\theta}$.

Bayesian strategy: Bayesian inference will require a probability model $y_i \sim p_\theta = P(y_i \mid \theta)$. So the model gives us $P(y_i \mid \theta)$, which will eventually give us $P(\mathbf{y} \mid \theta)$. We just need to switch it to get $P(\theta \mid \mathbf{y})$. Interestingly, this strategy of inference is due to Laplace but not Thomas Bayes. Hence, we get a 'distribution' of θ as a function of the data $\mathbf{y}$ instead of getting an estimation formula for θ as a function of the data $\mathbf{y}$ like $\delta(\mathbf{y})$ in a frequentist setting.

But how to switch? Bayes theorem is at answer. This theorem states that $P(A \mid B)P(B) = P(B \mid A)P(A)$. Thus in order to get $P(\theta \mid \mathbf{y})$ out of $P(\mathbf{y} \mid \theta)$, we can simply apply Bayes theorem and get $P(\theta \mid \mathbf{y}) = \frac{P(\theta)}{P(\mathbf{y})} P(\mathbf{y} \mid \theta)$.

Thus, we need this extra term, $P(\theta)$ which is the so-called prior probability for θ. Several classes will be dedicated to this.

Hence, Bayesian methods are all Likelihood based methods, and they follow the two Likelihood principals as (Berger and Wolpert 1984). Fun fact: While writing down these likelihood principals for the famous "Casella-Berger" inference book, Berger became a Bayesian.

The likelihood principal: Inference based on the likelihood function naturally adheres to two likelihood principles (LP) (Berger and Wolpert 1984):

1. Likelihood principle 1: All evidence, which is obtained from an experiment, about an unknown quantity θ is contained in the likelihood function of θ for the given data.

2. Likelihood principle 2: Two likelihood functions for θ contain the same information about θ if they are proportional to each other.

His reason was: Even when a Frequentist approaches a problem with a Likelihood i.e. assumes a probability model for the data (usually for non-continuous data like binary, count etc, a probability model is necessary), the inference strategy may not always Likelihood principals. Like example, I.7 in the book.

There is another school of thought which is less popular, called Fiducial inference. R.A. Fisher introduced this idea to bridge the gap between Frequentist and Bayesian. Later, this became obsolete due to some fundamental issues. But recently, statisticians rejuvenated this approach again by introducing "Generalized Fiducial inference" (Look it up this paper [1]).

References

[1] Jan Hannig, Hari Iyer, Randy CS Lai, and Thomas CM Lee. Generalized fiducial inference: A review and new results. *Journal of the American Statistical Association*, 111(515):1346–1361, 2016.

[2] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[3] Emmanuel Lesaffre and Andrew B Lawson. *Bayesian biostatistics*. John Wiley & Sons, 2012.

Part 2

The specification of prior probability may correspond to a "limiting proportion that an event happens in a true or fictive experiment". Like prior probability of head while tossing coin. This is called objective prior probability. (It is often driven by the motivation to simplify the calculations.)

Or it may come from a "a personal belief/experience/historical data". It is called subjective prior probability. Such kind of prior is often "constructed" based on "qualitative" prior belief.

Probability can be viewed as a function from the sample space to a value between $[0, 1]$. Sample space contains all possible events from an experiment. This function must satisfy a set of conditions such as:

In particular for mutually exclusive events $A_1, A_2, \ldots, A_K$ with the total event S (A_1 or A_2 or $\ldots$ or A_K):

- For each event A (from $A_1, A_2, ..., A_K$) : $0 \leq P(A) \leq 1$.
- The sum of all probabilities should be one: $P(S) = p(A_1 \text{ or } A_2 \text{ or } \cdots \text{ or } A_K) = 1$, and also $P(A_i \text{ or } A_j \text{ or } \cdots \text{ or } A_k) = P(A_i) + P(A_j) + \ldots + P(A_k)$.
- The probability that event A will not happen (event A^c) is $1 -$ the probability that A will happen: $P(A^c) = 1 - P(A)$.
- Suppose $B_1, B_2, \ldots, B_L$ represent another subdivision of S, then $P(A_i \mid B_j) = P(A_i \cap B_j)P(B_j)$, with $P(A_i \cap B_j)$ the probability that A_i and B_j happen together and $P(A_i \mid B_j)$ the conditional probability that A_i happens given that B_j has already happened.

Categorical version: Let the parameter θ in statistical problem take K possible values $\{\theta_1, \ldots, \theta_K\}$. Then

$$P(\theta_i \mid y) = \frac{P(y \mid \theta_i)P(\theta_i)}{\sum_k P(y \mid \theta_k)P(\theta_k)}$$

Continuous version: As we wrote last time $P(\theta \mid \mathbf{y}) = \frac{P(\theta)}{P(\mathbf{y})} P(\mathbf{y} \mid \theta)$. The denominator is the marginal distribution of the data, which we do not need to compute explicitly to write down the posterior. We also call $P(\mathbf{y}) = \int P(\mathbf{y}|\theta)P(\theta)d\theta$ is called *prior predictive distribution.* We can thus usually work with the proportional likelihood $P(\theta \mid \mathbf{y}) \propto P(\mathbf{y} \mid \theta)P(\theta)$. Technically, it is NOT the posterior probability. We do not need to compute $P(\mathbf{y})$ and take an alternative approach for posterior inference.

However, this above proportional representation will not be useful for the categorical case. We will see how.

Although, we are assuming the parameter θ to be stochastic (i.e. a random element, not fixed), Bayesians do indeed believe the existence of a true parameter θ_0. However, the prior distribution $P(\theta)$ essentially exploits the randomness in the parameter due to our limited resource of information. If somebody knows everything about θ, for him $P(\theta)$ will be degenerate distribution i.e. for him $P(\theta)$ will be positive only for one value i.e. true value of θ.

Another term for $P(\mathbf{y}|\theta)$ is likelihood which is denoted as $L(\theta \mid \mathbf{y})$. Thus $L(\theta \mid \mathbf{y}) = P(\mathbf{y} \mid \theta) = P(y_1, \ldots, y_n|\theta)$. If the data points are independent, then $P(y_1, \ldots, y_n \mid \theta) = P(y_1 \mid \theta)P(y_2 \mid \theta) \ldots P(y_n \mid \theta) = \prod_{i=1}^{n} P(y_i \mid \theta)$. Hence, for independent data points $L(\theta|\mathbf{y}) = \prod_{i=1}^{n} P(y_i \mid \theta)$. The formula for $P(y_i \mid \theta)$ depends on distributional assumptions on the data whether they are normal or Poisson or binomial etc.

The general strategy is to choose the prior distribution to make the computation easy. And set the parameters of this prior distribution based on some qualitative or historical knowledge on the parameter.

The posterior $P(\theta \mid \mathbf{y})$ may not always look like a standard well-known distribution. However, our aim is to have $P(\theta \mid \mathbf{y})$ to be from the standard list as a lot of properties are known about those. Additionally, only a very few number of parameters are needed for its characterization. For example, if $P(\theta \mid \mathbf{y})$ takes the form of Gaussian distribution, we can just need to get its mean and variance. For Poisson, it's just the mean, etc.

For a given type of likelihood function $L(\theta \mid \mathbf{y}) = P(\mathbf{y} \mid \theta)$, if there exists a class of distribution which can give us a proper distribution for $P(\theta|\mathbf{y})$, that should be our prior choice of distribution for $P(\theta)$.

Remark: Let us assume data are collected in two stages and θ be the parameter of interest. Let $\mathbf{D}_1$ and $\mathbf{D}_2$ stand for the data collected at stage 1 and stage 2, respectively. Now, if we use the likelihood from stage 1 as the prior for a Bayesian analysis of stage 2, the posterior mode = MLE of θ using the combined data $\{\mathbf{D}_1, \mathbf{D}_2\}$

Proof of the above statement: Likelihood of the complete data $L(\theta \mid \mathbf{D}_1, \mathbf{D}_2) = P(\mathbf{D}_1, \mathbf{D}_2 \mid \theta) = P(\mathbf{D}_1 \mid \theta)P(\mathbf{D}_2 \mid \theta)$, assuming independence of the two sets of data $\mathbf{D}_1$ and $\mathbf{D}_2$. Using likelihood of stage 1 as prior, we should have $P(\theta) \propto P(\mathbf{D}_1 \mid \theta)$. Then the posterior

$P(\theta \mid \mathbf{D}_2) \propto P(\mathbf{D}_2 \mid \theta)P(\theta) \propto P(\mathbf{D}_2 \mid \theta)P(\mathbf{D}_1 \mid \theta) = L(\theta \mid \mathbf{D}_1, \mathbf{D}_2)$. [Note that the proportionality implies that the ignored constants do not depend on θ. If there is any term involving θ, that should be written down explicitly.] Hence, posterior is proportional to the likelihood of the complete data. Hence, MLE of the complete data = mode of the posterior distribution.

Prior posterior and likelihood interplay: In Bayesian inference, we have two sources of information, the prior (source 1) and the likelihood (source 2). Intuitively, the posterior distribution is a combination of these two sources. We cannot manipulate source 2 as it is directly coming from the data. But we can modify source 1 as needed. If source 1 is very dominating, then the posterior will be aligned more towards the prior. However, if source 1 does not have any extra information, the posterior will be primarily guided by the likelihood only.

Putting a prior is like putting a soft-constraint: In the binomial example from the book, $\theta \in [0, 1]$. It can take any value between $[0, 1]$. But a researcher believes it is very unlikely that $\theta < 0.2$ or $\theta > 0.6$. Hence, a frequentist MLE estimation might consider employing a constraint optimization restricting $\theta \in [0.2, 0.6]$. But it is risky. There is a off chance the true parameter might be outside of this interval. Now due to the above interplay, we put a strong prior for θ such that $P(\theta \in [0.2, 0.6]) = 1 - \alpha$. As I decrease α, the prior will concentrate more more within this interval, compelling the posterior too to concentrate more in this interval. If the likelhood also suggest that the parameter is expected to be in this interval only, this extra effect of the prior will not impact much. But if the likelihood suggest otherwise, the prior will still give push so that the posterior gets a high concentration between $[0.2, 0.6]$.

Non-informative prior: Non-informative priors are those which exist to help us to proceed with a Bayesian inference without providing any extra information about the parameter. Like in the above Binomial case, if prior $P(\theta) = \text{Unif}(0, 1)$, the $P(\theta \mid y) \propto P(y \mid \theta) = L(\theta \mid y)$, hence the posterior is proportional to the likelihood. Hence, posterior distribution contain only the information provided by the likelihood and nothing more.

In general, non-informative priors should satisfy $P(\theta) \propto$ Constant. Now, note that for probability density to be *proper*, we must have $\int P(\theta)d\theta = 1$ (Definition). Now, if $P(\theta) \propto$ Constant defined in a bounded domain $[a, b]$, then $\int P(\theta)d\theta = \text{Constant}(b - a) = 1$. Hence, the Constant$=\frac{1}{b-a}$. Hence, if either of b and a is unrestricted (meaning $a \to -\infty$ or $b \to \infty$ or both), the density value of $P(\theta) = 0$ for all θ, which is not a desired density for a prior. Hence, for parameters supported in an unbounded domain, we work with *weakly informative* priors if we do not want to impose any strong prior belief as well as want to maintain reasonable posterior inference. These priors are also called *flat* priors.

Gaussian result (assuming the variance σ is known):

Intercept-only regression: Let $\mathbf{y}_i = \mu + \epsilon_i$, where $\epsilon_i \sim \text{Normal}(0, \sigma^2)$.

Here, the likelihood for a dataset $\mathbf{y} = (y_1, \ldots, y_n)$ that follows $y_i \sim \text{Normal}(\mu, \sigma^2)$ is $L(\mu \mid \mathbf{y}) = P(\mathbf{y} \mid \mu) = \prod_{i=1}^{n} P(y_i \mid \mu) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2}\frac{(\mu-\bar{y})^2}{\sigma^2/n}\right)$, where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The prior for $\mu \sim \text{Normal}(\mu_0, \sigma_0^2)$. Why are we choosing Normal distribution as the prior distri-

bution?: It is again to make the computation easy. With a normal prior + normal likelihood, the posterior becomes normal distribution again.

$$
\begin{aligned}
P(\mu \mid \mathbf{y}) \propto P(\mathbf{y} \mid \mu)P(\mu) &\propto \exp\left(-\frac{1}{2}\frac{(\mu-\bar{y})^2}{\sigma^2/n} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right) \\
&= \exp\left[-\frac{1}{2}\mu^2\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right) + \mu\left(\frac{n\bar{y}}{\sigma^2}+\frac{\mu_0}{\sigma_0^2}\right)\right] \\
&\propto \exp\left[-\frac{1}{2}\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\left(\mu - \frac{\frac{n\bar{y}}{\sigma^2}+\frac{\mu_0}{\sigma_0^2}}{\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}}\right)^2\right],
\end{aligned}
$$

which could be compared with $\exp\left(-\frac{(\mu-\bar{\mu})^2}{2\bar{\sigma}^2}\right)$ and obtain that $\bar{\sigma}^2 = 1/\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)$ and $\bar{\mu} = \frac{\frac{n\bar{y}}{\sigma^2}+\frac{\mu_0}{\sigma_0^2}}{\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}}$. Hence, the posterior distribution $P(\mu \mid \mathbf{y})$ follows a normal distribution with mean $\bar{\mu}$ and variance $\bar{\sigma}^2$.

Implications of above posterior:

- Any distribution can be specified entirely by their associated parameters. Looking at the posterior distribution, as $n \to \infty$ for a fixed μ_0 and σ_0, we have, $\bar{\mu} = \frac{\frac{\bar{y}}{\sigma^2}+\frac{\mu_0}{n\sigma_0^2}}{\frac{1}{\sigma^2}+\frac{1}{n\sigma_0^2}} \to \bar{y}$, the MLE estimate of μ taking a frequentist route.
- In the above scenario, we have $\bar{\sigma}^2 = 1/\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right) \to 0$. Hence, the posterior distribution approaches to Normal$(\bar{y}, 0)$, which is degenerate at $\bar{y}$.
- Similar to binomial case, let us assume that μ_0 and σ_0 are estimated using a historical data. If we are getting more and more historical data, it is expected that we achieve more and more certainty about μ. Hence, we may assume $\sigma_0 \to 0$. In that case, $\bar{\mu} = \frac{\frac{\bar{y}}{\sigma^2}+\frac{\mu_0}{n\sigma_0^2}}{\frac{1}{\sigma^2}+\frac{1}{n\sigma_0^2}} = \frac{\frac{\bar{y}\sigma_0^2}{\sigma^2}+\frac{\mu_0\sigma_0^2}{n}}{\frac{\sigma_0^2}{\sigma^2}+\frac{\sigma_0^2}{n}} \to \mu_0$ and $\bar{\sigma}^2$ again tends to zero. Thus, the posterior again concentrates more on the prior.

Weakly informative prior for Normal case: The support of μ is the entire real line. Hence, we can never find a proper distribution for the prior such that $P(\mu) \propto$ Constant. But in Figure 2, we see that the normal distribution tends to be *flatter* as we increase the variance in a given interval. Hence, it is possible to make the distribution $P(\mu)$ flat between a *pre-specified* interval, say $[a, b]$. Let us assume that we want the $P(\mu)$ to be flat. One possible solution is to first set $\mu_0 = \frac{a+b}{2}$. For $a = -1$ and $b = 1$, setting $\sigma_0 \geq (b-a) = 2$ will almost give a flat prior between

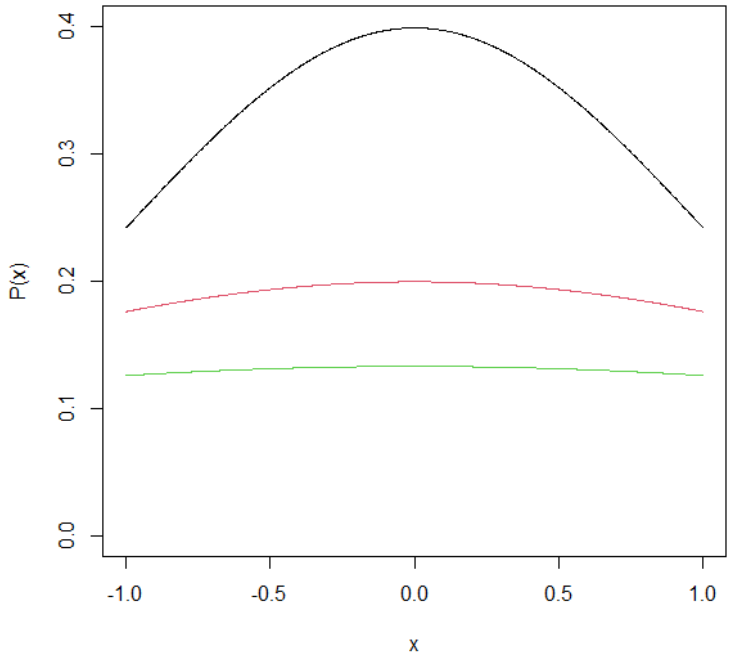

Figure 2: Black: Normal$(0, 1^2)$, Red: Normal$(0, 2^2)$ and Green: Normal$(0, 3^2)$.

$(-1, 1)$. Hence, we do not get flatness throughout the real, but only within a given interval which may serve the purpose. The choice $\sigma_0 \geq (b - a)$ will mostly hold.

For a normal likelihood and normal prior, the prior predictive distribution $P(\mathbf{y})$ is a normal distribution.

Most of the results for Poisson distribution will be similar to the normal case. The most *convenient* (conjugate) prior for a Poisson likelihood is the Gamma distribution as prior. Hence, the complete 'Bayesian' model is that $y_i \sim \text{Poisson}(\lambda)$ and $\lambda \sim \text{Ga}(a_0, b_0)$. [Bayesian model = Model + prior.] The prior predictive distribution $P(\mathbf{y})$ in this case is the negative binomial distribution. This is a very well-known result in statistics. Negative binomial is known to be a more flexible model for Count valued data than Poisson. With a conjugate Gamma prior on the mean parameter of Poisson, we achieve that immediately.

The weakly informative gamma prior is $\text{Ga}(c, c)$, where c very close to zero, like 0.001. [The book has wrongly stated that prior should be $\text{Ga}(1, c)$.]

Jacobian transformation: Let h be a monotone transformation of the parameter θ such that $\psi = h(\theta)$. Jacobian transformation holds both for the prior and posterior distribution of θ and thus the distribution of ψ is given by $P(h^{-1}(\psi)) \left(\mid \frac{d\psi}{d\theta} \mid \right)^{-1}$. Note that the first part of this expression when $P(h^{-1}(\psi))$ is actually $P(\theta)$, the (prior or posterior) distribution of θ.

The prior distribution provides an alternative source of information. It does not require coming "prior" to the data collection.

However, if the data is coming in batches, a Bayesian approach can easily incorporate that, as in example V.1 in the book. We have seen such example before too. We compute the posterior of study 1 and use it as prior for the study 2. In this situation, the prior distribution indeed comes from a historical source that was collected before the current data in hand.

But that is not the standard way. There are situations when we specify the prior after observing the data.

Conjugate family of priors: Please review the following Table 3 which is collected from [3]. The entry in the parameter column of Normal-mean fixed should be σ^2, NOT μ. We use all of these prior whenever possible to make the Bayesian computation easy.

Conditional conjugate and semiconjugate priors The prior for example $\mu \sim \text{Normal}(\mu_0, \sigma_0^2), \sigma^2 \sim \text{Inv} - \chi^2(\nu_0, \tau_0^2)$ is conditional conjugate. This is the case from 4.3.3 of the book. We noted that the sampling will be hard to perform for this prior. However, there are sampling schemes to tackle this kind of priors. Specifically for conditionally conjugate priors there are efficient samplers which will discuss later.

What is Conditional conjugate prior? Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ are three parameters for a statistical model. As an example, the data Y_i may follow a linear regression model with mean $\theta_1 X_{i,1} + \theta_2 X_{i,2}$

Table 5.2 Common members of the exponential family and their associated (natural) conjugate prior.

Exponential family member		Parameter	Conjugate prior
Univariate case			
Discrete distributions			
Bernoulli	Bern(θ)	θ	Beta(α_0, β_0)
Binomial	Bin(n, θ)	θ	Beta(α_0, β_0)
Negative binomial	NB(k, θ)	θ	Beta(α_0, β_0)
Poisson	Poisson(θ)	θ	Gamma(α_0, β_0)
Continuous distributions			
Normal-variance fixed	N(μ, σ^2)-σ^2 fixed	μ	N(μ_0, σ_0^2)
Normal-mean fixed	N(μ, σ^2)-μ fixed	μ	IG(α_0, β_0) Inv-χ^2(ν_0, τ_0^2)
Normal*	N(μ, σ^2)	μ, σ^2	NIG(μ_0, κ_0, a_0, b_0) N-Inv-χ^2(μ_0, κ_0, ν_0, τ_0^2)
Exponential	Exp(λ)	λ	Gamma(α_0, β_0)
Multivariate case			
Discrete distributions			
Multinomial	Mult(n, $\boldsymbol{\theta}$)	$\boldsymbol{\theta}$	Dirichlet($\boldsymbol{\alpha}_0$)
Continuous distributions			
Normal-covariance fixed	N($\boldsymbol{\mu}$, Σ)-Σ fixed	$\boldsymbol{\mu}$	N($\boldsymbol{\mu}_0$, Σ_0)
Normal-mean fixed	N($\boldsymbol{\mu}$, Σ)-$\boldsymbol{\mu}$ fixed	Σ	IW(Λ_0, ν_0)
Normal*	N($\boldsymbol{\mu}$, Σ)	$\boldsymbol{\mu}$, Σ	NIW($\boldsymbol{\mu}_0$, κ_0, ν_0, Λ_0)

Figure 3: Conjugate priors for different data distributions (Collected from [3])

and variance θ_3^2. Anyway, in this case, if we specify some conditionally conjugate priors for $\boldsymbol{\theta}$, It should satisfy two conditions.

1) The individual priors for θ_1, θ_2 and θ_3 should be independent, and the joint prior should be the product of the individual priors. Like the example in 4.3.3 the prior for (μ, σ) is $\text{Normal}(\mu_0, \sigma_0^2)\text{Inv} - \chi^2(\nu_0, \tau_0^2)$. Here $\text{Normal}(\mu_0, \sigma_0^2)$ and $\text{Inv} - \chi^2(\nu_0, \tau_0^2)$ are completely independent distributions.

2) These individual priors should be the conjugate choice, having all other parameters fixed. For example, the chosen prior for θ_2 should be a conjugate prior for a given statistical model if θ_1 and θ_3 are assumed completely known. Similar requirement would also hold for other parameters. Like in example 4.3.3, $\text{Normal}(\mu_0, \sigma_0^2)$ is a conjugate prior choice for μ when σ is fixed and known. On the other hand, $\text{Inv}-\chi^2(\nu_0, \tau_0^2)$ is a conjugate choice for σ when μ is fixed and known.

Non-informative prior Why non-informative prior? Because we want to a Bayesian analysis as it helps to quantify uncertainty from the posterior samples but we also do not want to make my analysis influenced by what prior I am using.

Our initial choices for non-informative priors were due to Bayes–Laplace postulate. But it has some issues.

When we talk about non-informative prior, we want the prior should exert the same information for all the values in the parameter space. Hence, we want $P(\theta)$ to be the same for all θ. This leads to issues when we consider transformed parameter. If $\theta \sim \text{Unif}(0, 1)$, we can show $P(\psi) = \exp(-\psi)$, which is an Exponential distribution with mean 1, where $\psi = -\log(\theta)$. This is using the Jacobian transformation. [Short proof without Jacobian: $P(\psi \leq a) = P(-\log(\theta) \leq a) = P(\theta \leq e^{-a}) = e^{-a}$, as $\theta \sim \text{Unif}(0, 1)$. e^{-a} is the cumulative distribution function for Exponential(1).] Jacobian transformed was only defined for monotone transformation, where the transformation function could either be increasing or decreasing. Like log is monotone increasing function.

Then, we definitely do not have $P(\psi)$ to be equal for all the possible values of ψ. Hence, non-informative prior on θ do not lead to a non-informative prior for ψ in the usual sense.

Jeffery's prior The non-informativeness in Jeffery's prior is different from the one discussed above. The relative probability assigned to a volume of a probability space using a Jeffrey's prior will be the same regardless of the parameterization used to define the Jeffrey's prior.

Say $\theta = f(\varphi)$, then we need $\int_{\varphi \in B} P_\varphi(\varphi) = \int_{\theta \in f(B)} P_\theta(\theta)$.

If θ and φ are two possible parametrizations of a statistical model, and θ is a continuously differentiable function of φ, we say that the prior $P_\theta(\theta)$ is "invariant" under a reparametrization if $P_\varphi(\varphi) = P_\theta(\theta) \left| \frac{d\theta}{d\varphi} \right|$ that is, if the priors $P_\theta(\theta)$ and $P_\varphi(\varphi)$ are related by the usual change of variables theorem.

Fisher information transforms under reparametrization as $I_\varphi(\varphi) = I_\theta(\theta) \left(\frac{d\theta}{d\varphi} \right)^2$ and thus

we can set $P_\varphi(\varphi) \propto \sqrt{I_\varphi(\varphi)}$ which will satisfy the change of variable condition. $I_\theta(\theta) = \mathbb{E}\{\left(\frac{d}{d\theta} f(\mathbf{y} \mid \theta)\right)^2 \mid \theta\}$, which is the expectation of the square of the score function (which is the first derivative of the likelihood).

For multiparameter $P_\varphi(\varphi) \propto \sqrt{det(I_\varphi(\varphi))}$, where *det* stands for the determinant. Thus, prior becomes the square-root of the determinant of the Fisher's info. The proportional sign is to denote that there is a constant. Constants of the prior are often ignored, as we can compute the constant of the posterior directly by checking its distributional properties.

There are several issues with Jeffery's prior. It's most often improper i.e. it does not lead to a valid density like for normal case it is $(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$ which is not a proper density as $\int\int \frac{1}{\sigma^2} d\sigma^2 d\mu = \infty$.

There is a philosophical issue is that Jeffery's prior violates the Likelihood principle as it is based on Fisher's information, which is an expectation over all possible values. For example binomial likelihood and negative-binomial likelihood will produce different Jefferey's priors, however, they have the same conjugate prior as these two likelihoods are proportionally equivalent. Two examples of Jeffrey's prior are given below.

Example 1 Let $y \sim \text{Normal}(\mu, \sigma^2)$. The log-likelihood $= C - \frac{np}{2}\log(\sigma^2) - \frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{2\sigma^2}$. Note that in this model, $\mathbb{E}(y-\mu) = 0$ and $\mathbb{E}(y-\mu)^T(y-\mu) = \sigma^2$.

Thus the Fisher information is, $\mathbf{I}(\boldsymbol{\beta}, \boldsymbol{\sigma^2}) = \mathbb{E}_{\mathbf{y}}\begin{bmatrix} \frac{1}{2}\frac{1}{(\sigma^2)^2} - \frac{(y-\mu)^T(y-\mu)}{4(\sigma^2)^3} & -\frac{(y-\mu)}{(\sigma^2)^2} \\ -\frac{(y-\mu)}{2(\sigma^2)^2} & -\frac{1}{2(\sigma^2)^2} \end{bmatrix} = (\sigma^2)^{-2}\begin{bmatrix} \frac{1}{2} - \frac{1}{4} & 0 \\ 0 & -1 \end{bmatrix}$.

Hence, Jeffrey's prior for μ, σ^2 is $P(\mu, \sigma^2) \propto \sqrt{det(I(\mu, \sigma^2))} \propto \sigma^{-2}$.

Example 2 Let $\mathbf{y} \sim \text{Normal}(\mathbf{X}\boldsymbol{\beta}, \sigma^2)$, where $\mathbf{X}$ is $n \times p$ design matrix. The log-likelihood $= C - \frac{np}{2}\log(\sigma^2) - \frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{2\sigma^2}$. Note that in this model, $\mathbb{E}(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$ and $\mathbb{E}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = n\sigma^2$

Thus the Fisher information is, $\mathbf{I}(\boldsymbol{\beta}, \boldsymbol{\sigma^2}) = \mathbb{E}_{\mathbf{y}}\begin{bmatrix} \frac{np}{2}\frac{1}{(\sigma^2)^2} - \frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{4(\sigma^2)^3} & -\frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T\mathbf{X}}{(\sigma^2)^2} \\ -\frac{\mathbf{X}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{(\sigma^2)^2} & -\frac{\mathbf{X}^T\mathbf{X}}{2(\sigma^2)^2} \end{bmatrix} = (\sigma^2)^{-2}\begin{bmatrix} \frac{np}{2} - \frac{1}{4} & 0 \\ 0 & -\mathbf{X}^T\mathbf{X} \end{bmatrix}$.

Hence, Jeffrey's prior for $\boldsymbol{\beta}, \sigma^2$ is $P(\boldsymbol{\beta}, \sigma^2) \propto \sqrt{det(I(\boldsymbol{\beta}, \sigma^2))} \propto (\sigma^{-2})^{\frac{p+2}{2}}$. We often ignore this power $\frac{p+2}{2}$ and set $P(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$ instead.

Reference prior: It is a non-informative prior that maximizes the Kullback–Leibler divergence between the prior and the posterior. Kullback–Leibler divergence quantifies the distance between two distributions. By taking the prior which maximizes the distance between the prior distribution and the posterior distribution essentially will lead to setting where the prior does not influence the posterior by a great lot.

Improper prior When the assumed prior is not a valid probability distribution, it is called an improper prior, like $(\mu, \sigma^2) \sim \frac{1}{\sigma^2}$. But the resulting posterior from an improper can be a valid posterior.

Weak/vague priors These are the priors when we chose a prior distribution from a valid distribution family and set the hyperparameters ensuring almost flatness at a pre-specified region. We have seen examples like Normal$(0, 3^2)$ is almost flat in the region $[-1, 1]$.

Informative prior This kind of prior distributions are due to historical data, historical data+expert knowledge, etc. Elicitation of prior information is an important area of research. It develops methods to optimally incorporate expert opinion into the prior distributions.

Part 3

In statistics, there are two types of summary estimates we consider, 1) Point estimate (sample estimates of the parameters) and 2) Interval estimate (Confidence interval). In Bayes inference, point estimates are like posterior mean, posterior median, posterior mode. Interval estimates are more general than frequentist ways as we can compute posterior probability of any interval.

Posterior point estimation

Posterior mode: Definition of posterior model is $\hat{\theta}_M = \arg\max_\theta P(\theta \mid \mathbf{y})$. 1) Hence, to get posterior mode, we just need to perform maximization of $P(\theta \mid \mathbf{y})$ or $L(\theta \mid \mathbf{y})P(\theta)$. 2) For a flat prior, we have $P(\theta) \propto C$, hence $P(\theta \mid \mathbf{y}) \propto L(\theta \mid \mathbf{y})$. So "Posterior mode with a flat prior = MLE". 3) A monotone transformation may not always preserve the mode. Or, the image of the posterior mode under a monotone transformation h is in general not a posterior mode anymore. In other words, $\hat{\psi}_M \neq h(\hat{\theta}_M)$ for $\psi = h(\theta)$. Simple example: If $\theta \sim$ Normal(μ, σ^2), then $\hat{\theta}_M = \mu$. For $\psi = \exp(\theta)$, i.e. ψ follows a log-normal distribution. Then $\hat{\psi}_M = \exp(\mu - \sigma^2) \neq \exp(\mu)$. It is happening due to the change in normalizing constant as a by-product of the Jacobian.

Posterior mean: Next is posterior mean $\bar{\theta} = \int \theta P(\theta \mid \mathbf{y}) d\theta$. 1) We can also write $\bar{\theta} = \arg\min_{\theta^*} \int (\theta - \theta^*)^2 P(\theta \mid \mathbf{y}) d\theta$. 2) The image of the posterior mean under a monotone transformation h is in general not a posterior mean anymore. Simple example: If $\theta \sim$ Normal(μ, σ^2), then $\bar{\theta} = \mu$. For $\psi = \exp(\theta)$, i.e. ψ follows a log-normal distribution. Then $\bar{\psi} = \exp(\mu + \sigma^2/2) \neq \exp(\mu)$.

Posterior median (univariate): Posterior median is $\bar{\theta}_M$ such that $P(\theta \leq \bar{\theta}_M \mid \mathbf{y}) = 0.5$. 1) We can also write $\bar{\theta} = \arg\min_{\theta^*} \int |\theta - \theta^*| P(\theta \mid \mathbf{y}) d\theta$. (quadratic loss is replaced by absolute loss.) 2) The image of a posterior median under a monotone transformation h is again a posterior median. 3) For a unimodal symmetric posterior distribution (such as normal, Beta(s, s) such that $s > 1$), the posterior median is equal to the posterior mean and equal to the posterior mode.

Bayesian inference is about drawing inferences of θ from $P(\theta \mid \mathbf{y})$. Now if the conditional distribution $P(\theta \mid \mathbf{y})$ belongs to any known class of probability distributions, you can directly use the summary statistics such as mean, median, mode, variance, etc. of that distribution. Some popular cases are: a) if $L(\theta \mid \mathbf{y}) = P(\mathbf{y}|\theta)$ is normal and $P(\theta)$ is also normal, we showed $P(\theta \mid \mathbf{y})$ is also normal, b) if $L(\theta \mid \mathbf{y}) = P(\mathbf{y}|\theta)$ is binomial and $P(\theta)$ is beta, we showed $P(\theta \mid \mathbf{y})$

is a beta distribution, c) if $L(\theta \mid \mathbf{y}) = P(\mathbf{y}|\theta)$ is Poisson and $P(\theta)$ is gamma, we showed $P(\theta \mid \mathbf{y})$ is a gamma distribution, etc.

Posterior interval estimation

Similar to frequentist scheme, we can define an interval estimate of a parameter θ as the range (most often the interval) of parameter values θ that are a posteriori most plausible with probability $(1-\alpha)$. Then, $[a, b]$ is a $100(1-\alpha)\%$ credible interval for θ if $P(a \leq \theta \leq b \mid \mathbf{y}) = 1-\alpha$. With $\alpha = 0.05$, the interval is called 95% credible interval or 95% CI (this is the most common interval). To get a feasible a and b, we need to solve $P(a \leq \theta \leq b \mid \mathbf{y}) = F(b) - F(a) = 1 - \alpha$, where F is the posterior cumulative distribution function

Evidently, the pair (a, b) satisfying the above equation will not be unique. To make them sort of unique, there are two versions. One is equal tail credible interval where, in addition to the above equation, we require $F(a) = \alpha/2$ and $1 - F(b) = \alpha/2$. So, essentially, $a = F^{-1}(\alpha/2)$ and $b = F^{-1}(1 - \alpha/2)$. Alternatively, we can use the `quantile` function of R to get these from the samples if F is not known. This interval may be inferentially problematic as we may have $P(\theta_1 \mid \mathbf{y}) < P(\theta_2 \mid \mathbf{y})$ such that $\theta_1 \in [a, b]$ and $\theta_2 \notin [a, b]$. It mainly appears for the multimodal or heavy distributions.

Next is the highest posterior density (HPD) interval: This is the interval $[a, b]$ such $P(a \leq \theta \leq b \mid \mathbf{y}) = 1 - \alpha$ and $P(\theta_1 \mid \mathbf{y}) \geq P(\theta_2 \mid \mathbf{y})$ such that $\theta_1 \in [a, b]$ and $\theta_2 \notin [a, b]$.

The HPD interval contains the values of θ that are a posteriori most plausible, i.e. $P(\theta \mid \mathbf{y})$ is higher for all θ' s inside the HPD interval than for values outside the interval. Note that the HPD interval explicitly needs a density, while the equal tail interval only needs the cdf. It is in general hard to compute HPD as it requires numerical optimization.

Important properties: 1) The $100(1-\alpha)\%$ HPD interval is the shortest interval such that $P(a \leq \theta \leq b \mid \mathbf{y}) = 1 - \alpha$. 2) The image of an equal tail CI under a monotone transformation h is again an equal tail CI, but this property does not hold in general for an HPD interval. 3) For a unimodal symmetric posterior distribution, the equal tail credible interval equals the corresponding HPD interval.

Prediction under Bayesian setting: In a frequentist setting, we characterize the predicted data as $\tilde{\mathbf{y}} \sim P(\tilde{\mathbf{y}} \mid \hat{\theta})$, where $\hat{\theta}$ is an estimate of the true parameter θ using the available data $\mathbf{y}$. Hence, it is a two stage approach. First estimate $\hat{\theta}$ and then sample from $P(\tilde{\mathbf{y}} \mid \hat{\theta})$. Thus, the uncertainty in prediction is underestimated. However, a Bayesian route makes the conditional probability argument behind the predictive distribution more transparent. Therefore, the distribution of $\tilde{\mathbf{y}}$ is, given the observed data $\mathbf{y}$, equal to $P(\tilde{\mathbf{y}} \mid \mathbf{y}) = \int P(\tilde{\mathbf{y}} \mid \theta) P(\theta \mid \mathbf{y}) d\theta$. It is called the posterior predictive distribution. In Bayesian setting, the effect of θ is integrated out, and thus uncertainty is well taken into account.

As we have already discussed that there are two possible ways in characterizing a Bayesian solution for θ, point estimation and interval estimation. Similar characterization is also avail-

able for the predictive distribution. We can compute mean, median or mode of the predictive distribution. Or we can present interval estimates which are called posterior predictive interval. More specifically, an interval $[a, b]$ is a $100(1 - alpha)\%$-posterior predictive interval (PPI) when $P(a \leq \tilde{\mathbf{y}} \leq b \mid \mathbf{y}) = 1 - \alpha$.

Gaussian case: Posterior predictive distribution is Gaussian.

Binomial case: Posterior predictive distribution is beta-binomial distribution.

Poisson case: Posterior predictive distribution is Negative-bionomial.

Bayesian CLT:

The model is $y|\theta \sim \text{Binomial}(n, \theta)$ and $\theta \sim \text{Beta}(1/2, 1/2)$. The exact posterior is $\theta|y \sim \text{Beta}(Y + 1/2, n + 1/2)$.

Posterior mode is $\hat{\theta} = \frac{Y-0.5}{n-1}$ and Fisher's info is $A/\hat{\theta}^2 + B/(1-\hat{\theta})^2$, where $A = Y - 0.5$ and $B = n - Y - 0.5$. Asymptotic variance is 1/Fisher's info.

Proof of Fisher's info result: Fisher's info$=-\frac{d^2(\log P(\theta|y))}{d\theta^2}_{\theta=\hat{\theta}}$. We have, $P(\theta \mid y) \propto \theta^A(1-\theta)^B$. So, $\frac{d(\log P(\theta|y))}{d\theta} = \frac{A}{\theta} - \frac{B}{1-\theta}$. Performing the differentiation one more time, we get $\frac{d^2(\log P(\theta|y))}{d\theta^2} = -\frac{A}{\theta^2} - \frac{B}{(1-\theta)^2}$.

```
theta <- seq(0.001,0.999,.001) # Grid of thetas for plotting
 Y     <- 2                    # The data
 n     <- 5

 # Compute the posterior mode and Fisher information matrix

 A         <- Y-0.5
 B         <- n-Y-0.5
 theta_MAP <- A/(A+B)
 Info      <- A/theta_MAP^2+B/(1-theta_MAP)^2

 # Plot the true and approximate posteriors

 post1 <- dbinom(Y,n,theta)*dbeta(theta,0.5,0.5)
 post1 <- post1/sum(post1)
 post2 <- dnorm(theta,theta_MAP,sqrt(1/Info))
 post2 <- post2/sum(post2)

 plot(theta,post1,type="l",lwd=2,
      xlab=expression(theta),ylab="Posterior")
 abline(v=theta_MAP,col=3,lwd=2)
 lines(theta,post2,col=2,lwd=2)
 legend("topright",c("Exact","CLT","MAP"),bty="n",col=1:3,cex=1.5,lwd=2)
```

Learn a distribution by sampling

If $P(\theta \mid \mathbf{y})$ does not follow any known parametric distribution family. We then approximately learn the distribution $P(\theta \mid \mathbf{y})$ using samples $\theta_1, \ldots, \theta_K$ such that $\theta_i \sim P(\theta \mid \mathbf{y})$. To see that this is indeed valid.

```
theta <- rbeta(10000, 19, 133)
#Density of log(\theta) using Jacobian is given below
#densijacobian <- (1/beta(19,133)) * exp(log(theta))^19 * (1-exp(log(theta)))^(133-1)
#Use that to compute density values in following grid
thetagrid <- (1:1000)/1000#seq(range(theta)[1], range(theta)[2], length.out = 1000)
densijacobian <- (1/beta(19,133)) * exp(log(thetagrid))^19 * (1-exp(log(thetagrid)))^(133-1)
plot(density(log(theta)), col=1, type = 'l') #ploting density using Monte Carlo method,
#just transformed the generated data in the
#first line and computed numeric desity
points(log(thetagrid), densijacobian, col=2, type = 'l') #Jacobian computed densities
#Some standard distributions:
plot(density(theta), col=1, type = 'l') #ploting density using Monte Carlo method
#(computed numeric density)
points(thetagrid, dbeta(thetagrid, 19, 133), col=2, type = 'l') #r function computed density
x <- rnorm(1000, 0, 1)
xgrid <- seq(-3,3,length.out = 1000)
plot(density(x), col=1, type = 'l') #ploting density using Monte Carlo method
#(computed numeric density)
points(xgrid, dnorm(xgrid, 0, 1), col=2, type = 'l') #r function computed density
```

Above examples show numerically computed densities from samples are matching with exact densities.

Then we need to learn sampling techniques for "non-standard" probability distributions. By non-standard, I mean the ones for which there does not exist any R program to sample θ directly. For example, if $L(\theta \mid \mathbf{y}) = P(\mathbf{y} \mid \theta)$ is Poisson and $P(\theta)$ is gamma, we showed $P(\theta \mid \mathbf{y})$ is a gamma distribution. So in this case, we can sample θ directly using `rgamma` in R. However, if the prior $P(\theta)$ is changed to log-normal, we cannot use any standard functions.

Numerical integration: Our focus is to integrate $f(x)$ with in $[a, b]$ i.e. we want to approximately compute $F(x) = \int_a^b f(x)$. The basic architecture of all the numerical integration methods is to first break the interval $[a, b]$ into several small intervals (usually of equal size). Let $a_1, a_2, \ldots, a_{M+1}$ are $M + 1$ break points such that $a_1 = a$ and $a_{M+1} = b$ with $a_{i+1} - a_i = \delta$ for all $i = 1, \ldots, M$. Then $\delta M = b - a$. Thus, $\delta = \frac{b-a}{M}$.

Firstly, $F(x) = \int_a^b f(x) = \sum_{i=1}^{M} \int_{a_i}^{a_{i+1}} f(x)$. We approximate each component $\int_{a_i}^{a_{i+1}} f(x)$ of the sum. Different methods replace $f(x)$ with different approximations. Different methods have different approximation errors.

Piece-wise constant or Riemann sum: In this case, the functional value is kept constant at $f(a_i)$ or $f(a_{i+1})$ in each of the intervals $[a_i, a_{i+1}]$. If it is the first one then $F(x) \approx \sum_{i=1}^{M} \delta f(a_i)$.

Trapezoidal rule: In this case, the functional value is kept constant at $\{f(a_i) + f(a_{i+1})\}/2$ in each of the intervals $[a_i, a_{i+1}]$. If it is the first one, then $F(x) \approx \sum_{i=1}^{M} \delta \frac{f(a_i)+f(a_{i+1})}{2}$.

Simpson's 1/3-rd rule: In this case, the functional value is kept constant at $\{f(a_i)+f(a_{i+1})\}/2$ in each of the intervals $[a_i, a_{i+1}]$. If it is the first one, then $F(x) \approx \sum_{i=1}^{M} \delta \frac{f(a_i)+4f(\frac{a_i+a_{i+1}}{2})+f(a_{i+1})}{6}$.

Laplace approximation: This method is used to approximate integrals of the form $\int_a^b e^{Mf(x)}dx$. It's approximation using Laplace's method is given by, $\sqrt{\frac{2\pi}{M|f''(x_0)|}}e^{Mf(x_0)}$ where x_0 is the global maxima of $f(x)$ between (a, b).

Proof for the simpler case is very straightforward. Since, x_0 is the global maxima of $f(x)$ between (a, b), we must have $f'(x_0) = 0$. By Taylor series expansion of $f(x)$ around x_0 is $f(x) \approx f(x_0) + f'(x_0)(x - x_0) - \frac{1}{2}|f''(x_0)|(x - x_0)^2 = f(x_0) - \frac{1}{2}|f''(x_0)|(x - x_0)^2$.
We can integrate $\int \exp(-\frac{1}{2}|f''(x_0)|(x - x_0)^2)dx$ by identifying its similarities with Gaussian density (specifically mean x_0 and variance $1/|f"(x_0)|$). This completes the proof.

Integration by sampling: To motivate integration by sampling, we can go back to our method of moment estimate. We know sample mean is an unbiased estimate of the population mean. Population mean is actually represented by an integration $\int xf(x)dx$, where x is a random variable with probability density $f(x)$. An estimate of that can be obtained by sample mean. Let $x_1, \ldots, x_n$ are n samples of x. Then we can approximate $\int xf(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} x_i$.

In case of normal distribution, we know $\int x\frac{1}{\sqrt{2\pi}\sigma}\exp(-(x - \mu)^2/(2\sigma^2)) = \mu \approx \frac{1}{n}\sum_{i=1}^{n} x_i$, where $x_1, \ldots, x_n$ are samples from Normal(μ, σ^2).

The above approximation holds due to the Strong Law of Large Numbers. We can even compute the approximation errors using Central limit theorem (CLT), which is called confidence interval in the context of parameter estimation. So, the confidence interval can be interpreted as the amount of error in the approximation.

Monte Carlo integration In the context of Bayes inference, we may often want to compute $E(t(\theta) \mid \mathbf{y})$. Here $t(\theta)$ is any function of θ such as $t(\theta) = \theta^2$ or $t(\theta) = 1/\theta$, etc. To approximate $\int t(\theta)P(\theta \mid \mathbf{y})d\theta$, we can draw samples of θ from posterior $P(\theta \mid \mathbf{y})$. Let $\theta_1, \ldots, \theta_n$ are n samples of θ from posterior $P(\theta \mid \mathbf{y})$. Then for n large enough, $\int t(\theta)P(\theta \mid \mathbf{y})d\theta \approx \frac{1}{n}\sum_{i=1}^{n} t(\theta_i)$.

Using CLT, for large n, we have $\frac{1}{n}\sum_{i=1}^{n} t(\theta_i) \to$ Normal$(E(t(\theta) \mid \mathbf{y}), \frac{s_n^2}{n})$, where s_n^2 is the variance of the sampled $t(\theta_i)$'s. $\frac{s_n}{\sqrt{n}}$ is called the Monte Carlo error. This error measures precision of the estimation procedure.

How to sample?

The inverse CDF (ICDF) method: This is only applicable for univariate method.And this is also an immediate application of what we have just seen above in the first example under "Learn a distribution by sampling". Let $F(x)$ is the cumulative distribution function (CDF) of

a random variable, X i.e. $F(x) = P(X \leq x)$ and we want to draw samples of X. We know that $F(X) \sim \text{Unif}(0,1)$. (Why? $P(F(X) \leq u) = P(X \leq F^{-1}(u)) = F(F^{-1}(u)) = u$. Thus $F(X) \sim \text{Unif}(0,1)$.) So we draw samples from uniform distribution which will be samples of $F(X)$ and then do F^{-1} transformation on these samples and get samples of X.

```
betasam1 <- rbeta(10000, 8, 10)

plot(density(betasam1), col=1, type = "l") #ploting empirical (empirical means computed from samples)
                                           #density using samples for Beta(8,10)
                                           #where samples are generated using R function rbeta

#Using ICDF
unifsam <- runif(10000)
betasam2 <- qbeta(unifsam, 8, 10)

points(density(betasam2), col=2, type = "l") #ploting empirical density using samples
                                           # for Beta(8,10) where samples are generated
                                           # using ICDF method

thetagrid <- (1:10000)/10000
points(thetagrid, dbeta(thetagrid, 8, 10), col=3, type = "l") #actual density
```

Accept-reject: The previous method requires one to know, F^{-1} which is a very strong requirement. For any nonstandard distribution, this will not be known anyway. Accept/reject is one way to sample in such scenarios. It can be implemented using R package `AR`

```
library("AR")
n <- 10 #number of trials
lambda <- 5
y <- rpois(10, lambda)  #number of success
lambdagivenY <- function(lambda){
  likelihood <- prod(dpois(y, lambda))
  prior      <- dnorm(lambda, mean = 5, sd = 1)

  out <- likelihood * prior
  return(out)
}
#lambdagivenY(lambda) is upper bounded by lambda^{sum(y)}*e^(-n*lambda) ignoring
#constant. A good choice for instrument is Beta(y+1, n-y+1)

samples <- AR.Sim(n=3000, lambdagivenY, Y.dist = "gamma", Y.dist.par = c(sum(y)+1,n))
plot(density(samples))


######Importance sample#####
J <- 1000000
K <- 3000 #Need to be K << J, K is the number of posterior samples you finally want
```

```
#Sample from the instrument distribution
gammasamples <- rgamma(J, sum(y)+1,n)

#Importance weights
weights      <- unlist(lapply(gammasamples, lambdagivenY))/dgamma(gammasamples, sum(y)+1,n)
weights <- weights/sum(weights)

#Generate K posterior samples using Importance sampling
postsamplesIS <- gammasamples[sample(1:J, K, prob = weights)]

points(density(postsamplesIS), col=2, type = 'l')
```

The basic algorithm: In the accept–reject (AR) algorithm, one first samples from an instrumental distribution $q(\theta)$. In a second step, some of the sampled values are rejected to end up with a sample from $P(\theta \mid \mathbf{y})$. The distribution $q(\theta)$ is called the proposal distribution and $P(\theta \mid \mathbf{y})$, in this context, the target distribution. The AR algorithm assumes that $P(\theta \mid \mathbf{y})$ is bounded above by a multiple of $q(\theta)$, i.e. there is a constant $A < \infty$ such that $P(\theta \mid \mathbf{y}) < Aq(\theta)$ for all θ. Therefore, the distribution q is also called the envelope distribution, A is the envelope constant and $Aq(\theta)$ is the envelope function. Sampling proceeds in two stages. In the first stage, a θ is drawn from $q(\theta)$ independently of u that is drawn from U(0, 1). In the second stage, θ is either accepted or rejected according to the following rule:

- Accept: When $u \leq P(\theta \mid \mathbf{y})/Aq(\theta)$, θ is accepted as a value from $P(\theta \mid \mathbf{y})$.
- Reject: When $u > P(\theta \mid \mathbf{y})/Aq(\theta), \theta$ is rejected.

Samples from Accept/Reject and Importance sampling show the same numeric density. (I will provide one example on ARS using R package `Runuran`. There were some issues. ARS is very difficult to handle in general)

Importance sampling: It was originally proposed to obtain simulation consistent estimates. If you have sampled $\theta_1, \ldots, \theta_K$ from $q(\theta)$ instead of the original posterior $P(\theta \mid \mathbf{y})$, then an approximate estimate $E(t(\theta \mid \mathbf{y})) \approx \sum_{k=1}^{K} t(\theta_k) \frac{w_k}{\sum_k w_k}$, where $w_k = \frac{P(\theta_k|\mathbf{y})}{q(\theta_k)}$.

The specific steps are:

- First stage: Draw J (with J large) independent values $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_J\}$ from $q(\theta)$ and calculate weights $w_j = w(\theta_j)(j = 1, \ldots, J)$ as above. This defines a multinomial distribution, with categories defined by the sampled θ values and associated probabilities $\mathbf{w} = (w_1, w_2, \ldots, w_J)$.
- Second stage: Take a sample of size K J, from $\boldsymbol{\zeta}$ i.e. draw from $Mult(K, \mathbf{w})$.

Then the drawn samples are $\boldsymbol{\theta}[\boldsymbol{\zeta}]$.

Hypothesis testing

Posterior credible interval approach where it is checked that where the credible interval contains the null hypothesis specified parameter or not.

Another approach is using Bayes factor. Let H_0 is our null hypothesis.

Bayes factor for null to alternative is defined as $\text{BF}_{01}(\mathbf{y}) = \frac{P(\mathbf{y}|H_0)}{P(\mathbf{y}|H_1)}$. Let us assume null is indeed true. Then your Bayes factor computation method is consistent if BF_{01} with increasing sample size i.e. the data vector $\mathbf{y}$ has more number of observations.

Then posterior odds for hypothesis H_0 = Bayes factor $\times$ prior odds for hypothesis H_0. "prior odds for hypothesis H_0" is kind of a bias that a Bayesian investigator would introduce in the model for the Bayesian route of inference. Now the motivation is to check whether posterior odds for hypothesis $H_0 \gtreqless$prior odds for hypothesis H_0. Thus, higher value of Bayes factor provides greater support to the null.

Jeffreys (1961, p. 432) classified the Bayes factor (favoring H_0 against H_a) into: 'decisive' ($\text{BF}(\mathbf{y}) > 100$), 'very strong' ($32 < \text{BF}(\mathbf{y}) \leq 100$), 'strong' ($10 < \text{BF}(\mathbf{y}) \leq 32$), 'substantial' ($3.2 < \text{BF}(\mathbf{y}) \leq 10$) and 'not worth more than a bare mention' ($1 < \text{BF}(\mathbf{y}) \leq 3.2$). These intervals may look artificial.

In the definition of Bayes factor, the expression is non-Bayesian. It's largely a likelihood ratio in the case of simple hypotheses. The Bayesian component is added while comparing complex hypotheses. Complex hypotheses are those which are specified by some interval. Check Example III.15

We can write $P(\mathbf{y}|H_0) = \int P(\mathbf{y}|\theta, H_0)P(\theta|H_0)d\theta$. Here $P(\mathbf{y}|\theta, H_0)$ is the likelihood under null hypothesis and $P(\theta|H_0)$ is the prior probability. Given H_0 part essentially put constraints on θ. For example, if $H_0 : \theta \leq 0$, then $P(\theta|H_0)$ ensures that the prior is only supported in the range $(-\infty, 0]$. Similarly, $P(\mathbf{y}|H_1) = \int P(\mathbf{y}|\theta, H_1)P(\theta|H_1)d\theta$.

Bayes factor requires us to compute $P(\mathbf{y}|H_0)$ and $P(\mathbf{y}|H_1)$ which are difficult for complex Bayesian model. Approximate methods are available.

First one is the Monte Carlo method where you draw sample $\{\theta_{1,0}, \ldots, \theta_{K,0}\}$ from the prior $P(\theta|H_0)$ and draw $\{\theta_{1,1}, \ldots, \theta_{K,1}\}$ from $P(\theta|H_1)$. Then compute the likelihood for each sampled θ and take the average $\frac{1}{K}\sum_{i=1}^{K} P(\mathbf{y}|\theta_{i,j})$ as your estimate of $P(\mathbf{y}|H_j)$ for $j = 0, 1$. This estimate is not very good.

Another option is using the Harmonic Mean Identity, which is more efficient than above Monte Carlo method. The identity essentially comes from importance sampling estimates of $P(\mathbf{y}|H_0)$ and $P(\mathbf{y}|H_1)$ based on the posterior samples of the underlying parameters. In this method, $P(\mathbf{y}|H_j) = \left(\sum_{i=1}^{K} \frac{P(\mathbf{y}|\theta_{i,j})^{-1}}{K}\right)^{-1}$ where $\theta_{1,j}, \ldots, \theta_{K,j}$ are sampled from the posterior distribution $P(\theta|\mathbf{y}, H_j)$. The only difference lies in how samples of θ are generated. When sampled from posterior, it is more efficient as it used information from the data [2].

References

[1] Jan Hannig, Hari Iyer, Randy CS Lai, and Thomas CM Lee. Generalized fiducial inference: A review and new results. *Journal of the American Statistical Association*, 111(515):1346–1361, 2016.

[2] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[3] Emmanuel Lesaffre and Andrew B Lawson. *Bayesian biostatistics*. John Wiley & Sons, 2012.

Part 4

Important messages from last chapters:

- Bayes is a likelihood based inference method, where we specify the mathematical characterization for a dataset through a probability distribution.
- Sampling is an important component for Bayesian computation.
- Sampling is more powerful than any optimization based inference methods as it captures more information of the estimate. (Digression: In frequentist framework too, resampling based bootstrap methods are getting increasingly popular for similar reasons. Using bootstrap methods, we can obtain empirical distribution of the parameter by resampling the original data.)
- The power of sampling is a bit hard to fully understand.
- Now, prior distribution can be used to introduce different types of information into the model. We can set a prior distribution which could be 1) Non-informative, or 2) Based on historical data, or 3) Informative prior (based on expert knowledge).
- As we move to more complex situations, you need to have better clarity on "proportional posterior". It is not the actual posterior. For many purposes, such as to identify the posterior distribution "proportional posterior" is sufficient as it only ignores the terms that do not have the parameters of interest.

Method of composition: $P(A, B, C) = P(A \mid B, C)P(B \mid C)P(C)$ or more generally, $P(A_1, A_2, \ldots, A_K) = P(A_1 \mid A_2, \ldots, A_K)P(A_2 \mid A_3, \ldots, A_K) \ldots P(A_K)$. However, this decomposition is not unique. We can follow any sequence of A_i's while writing down the decomposition. For example, $P(A, B, C) = P(A \mid B, C)P(B \mid C)P(C) = P(A \mid B, C)P(C \mid B)P(B) = P(B \mid A, C)P(A \mid C)P(C) = P(B \mid A, C)P(C \mid A)P(B) = P(C \mid A, B)P(A \mid B)P(B) = P(C \mid A, B)P(B \mid A)P(A)$.

If there are more than one parameter, you need to draw joint samples for inference. Say θ_1 and θ_2 are two parameters. Examples include normal distribution with mean and sigma both

unknown, generalized linear models with at least 2 predictors. We use method of composition to write 1) $P(\theta_1, \theta_2 \mid \mathbf{y}) = P(\theta_1 \mid \theta_2, \mathbf{y})P(\theta_2 \mid \mathbf{y})$ or 2) $P(\theta_1, \theta_2 \mid \mathbf{y}) = P(\theta_2 \mid \theta_1, \mathbf{y})P(\theta_1 \mid \mathbf{y})$.

Which of the above two possible decomposition between 1) and 2) to be selected for sampling will depend on which of the following two integrals is easy to do.

In the above decomposition $P(\theta_2 \mid \mathbf{y}) = \int P(\theta_1, \theta_2 \mid \mathbf{y})d\theta_1$, $P(\theta_1 \mid \mathbf{y}) = \int P(\theta_1, \theta_2 \mid \mathbf{y})d\theta_2$.

If it is easy to integrate both of the two integrals, we can draw samples of θ_1 and θ_2 completely independently from $P(\theta_1 \mid \mathbf{y})$ and $P(\theta_2 \mid \mathbf{y})$ respectively.

However, that's not always possible. The reason behind this decomposition is sampling. To draw a joint posterior sample of (θ_1, θ_2) from $P(\theta_1, \theta_2 \mid \mathbf{y})$ we can first draw a sample of θ_1 from $P(\theta_1 \mid \mathbf{y})$, which is called "marginal" posterior distribution of θ_1. Then we draw our sample of θ_2 from the "conditional" posterior $P(\theta_2 \mid \theta_1, \mathbf{y})$. The term "posterior" automatically implies 'conditional' on $\mathbf{y}$. Thus, $P(\theta_1 \mid \mathbf{y})$ is called marginal posterior (no 'conditional' terminology is used).

When do we need to compute marginal posterior for all the parameters? Like both $P(\theta_1 \mid \mathbf{y})$ and $P(\theta_2 \mid \mathbf{y})$: One scenario is when you want compute credible interval analytically by solving equations like $P(\theta_1 \leq a \mid \mathbf{y}) = \alpha/2$ and $P(\theta_1 \leq b \mid \mathbf{y}) = 1 - \alpha/2$. You need to know marginal posterior for all the parameters. If you compute these quantities from posterior samples, you do not need marginals for all but one.

Let's start with the most popular distribution, which is the Normal distribution with unknown mean (μ) and variance (σ^2).

In the Serum alkaline phosphatase study from Example III.6 of the book, Topal et al. (2003) measured serum alkaline phosphatase (alp) on a prospective set of 250 'healthy' patients. It was found that $y_i = 100/\sqrt{alp_i}, \quad (i = 1, ..., 250)$ has approximately a Gaussian distribution. While describing this example, $\sigma = 1.4$ was assumed on page 54 of the book.

But in reality, variance would never be known. Thus, it also needs to be estimated. In this chapter, we will study ways to estimate/sample multiple parameters simultaneously.

The sampling of multiple parameters has similarity with generating a predictive sample.

Non-informative Jeffrey's prior Just take it for granted that the non-informative prior for (μ, σ) is $P(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$. Few things: 1) This belongs to Jefferey's class of priors (Next chapter), and 2) This is not a proper probability density.

For a proper probability density, it must integrate to 1. Hence, we should have a proportional density to be integrated to something finite. Explanation: if $P(x) \propto c(x)$, then to ensure $\int P(x)dx = 1$, we need $\int c(x)dx = C < \infty$. Then we can set $P(x) = \frac{c(x)}{C}$.

However for the above prior $\int_{-\infty}^{\infty}\int_{0}^{\infty}\frac{1}{\sigma^2}d\sigma^2 d\mu = \infty$. Hence, $P(\mu,\sigma)$ can never be a proper unless we truncate the range of μ

Note that, in this proportional posterior $\frac{1}{2\sigma^2}(n-1)s^2$ is kept unlike the proportional expressions used in Equation 2.12 of the book. The reason is that in chapter 2, variance σ was given. It was not of interest, unlike this chapter, where σ is also a parameter of interest. Hence, our expression for proportional changes depending on the problem.

We can write $P(\mu,\sigma^2 \mid \mathbf{y}) = P(\mu \mid \sigma^2,\mathbf{y})P(\sigma^2 \mid \mathbf{y})$. Then marginals will $P(\mu \mid \mathbf{y}) = \int P(\mu,\sigma^2 \mid \mathbf{y})d\sigma^2$ and similarly $P(\sigma^2 \mid \mathbf{y}) = \int P(\mu,\sigma^2 \mid \mathbf{y})d\mu$.

Some definition: $P(\mu \mid \sigma^2,\mathbf{y})$ is called the conditional posterior and $P(\mu \mid \mathbf{y})$ is the marginal posterior.

The conditional posterior $P(\mu \mid \sigma^2,\mathbf{y})$ assumes σ^2 to be known. Hence, our previous derivations assuming σ^2 to be known will hold for this situation. Thus, $P(\mu \mid \sigma^2,\mathbf{y})$ will be same as Equation 2.19 from the book that $P(\mu \mid \sigma^2,\mathbf{y})$ is Normal$(\bar{\mathbf{y}},\sigma^2/n)$, which was the case where prior variance σ_0^2 of μ was tending to ∞. Above-mentioned Jeffrey's prior also puts equal 'mass' for all the values of μ, as in $P(\mu_1,\sigma^2) = P(\mu_2,\sigma^2)$ as long as σ part does not change. If the prior for μ is $P(\mu) = \text{Normal}(\mu_0,\sigma_0^2)$, then as $\sigma_0^2 \to 0$, then $P(\mu_1) \approx P(\mu_2)$. Hence, as long as σ^2 is known or given, the above two priors are equivalent.

Derivation for the marginal $P(\mu \mid \mathbf{y})$ being a t-distribution is a bit complicated.

Derivation for marginal of $P(\sigma^2 \mid \mathbf{y})$ is a bit simpler.

We rewrite Equation 4.4 as $\frac{1}{\sigma}\exp(-\frac{1}{2\sigma^2}n(\bar{\mathbf{y}}-\mu)^2) \times (\frac{1}{\sigma^2})^{\frac{n+1}{2}}\exp(-\frac{1}{2\sigma^2}(n-1)s^2)$.

Since $P(\mu \mid \sigma^2,\mathbf{y})$ is Normal$(\bar{\mathbf{y}},\sigma^2/n)$ and we have Normal$(\bar{\mathbf{y}},\sigma^2/n) = \frac{\sqrt{n}}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}n(\bar{\mathbf{y}}-\mu)^2)$. Thus, $\int \frac{\sqrt{n}}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}n(\bar{\mathbf{y}}-\mu)^2)d\mu = 1$, because Normal$(\bar{\mathbf{y}},\sigma^2/n)$ is a proper probability density.

So

$$\begin{aligned}
P(\sigma^2 \mid \mathbf{y}) &= \int \frac{1}{\sigma}\exp(-\frac{1}{2\sigma^2}n(\bar{\mathbf{y}}-\mu)^2) \times (\frac{1}{\sigma^2})^{\frac{n+1}{2}}\exp(-\frac{1}{2\sigma^2}(n-1)s^2)d\mu \\
&= (\frac{1}{\sigma^2})^{\frac{n+1}{2}}\exp(-\frac{1}{2\sigma^2}(n-1)s^2) \times \int \frac{1}{\sigma}\exp(-\frac{1}{2\sigma^2}n(\bar{\mathbf{y}}-\mu)^2)d\mu \\
&\propto (\frac{1}{\sigma^2})^{\frac{n+1}{2}}\exp(-\frac{1}{2\sigma^2}(n-1)s^2) \quad \text{(Verify using above integral result)} \\
&= (\frac{1}{\sigma^2})^{-\frac{n-1}{2}-1}\exp(-\frac{1}{2\sigma^2}(n-1)s^2),
\end{aligned}$$

which is the density for inverse-chiquared distribution for $\sigma^2/((n-1)s^2)$ with parameter $n-1$ or inverse-gamma distribution with parameters $(\frac{n-1}{2}, \frac{(n-1)s^2}{2})$. Inverse-chiquared distribution is

a special case of Inverse-gamma distribution.

Inverse-chisquared distribution: $P(x) = \frac{2^{-\nu/2}}{\Gamma(\nu/2)} x^{-\nu/2-1} e^{-1/(2x)} \propto x^{-\nu/2-1} e^{-1/(2x)}$
Inverse-gamma distribution: $P(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-\frac{\beta}{x}) \propto x^{-\alpha-1} \exp(-\frac{\beta}{x})$

So, setting $\alpha = \nu/2$ and $\beta = 1/2$, we get inverse-chisquared distribution. These two distributions actually only appear in Bayes inference.

One way to derive the marginal posterior of μ being a t-distribution result is to use the Normal-Inverse-Gamma mixture result. It just involves some hard integral steps. Note that $P(\sigma^2 \mid \mu, \mathbf{y})$ is an inverse-gamma with parameters $n/2$ and $\frac{(n-1)s^2 + n(\bar{\mathbf{y}}-\mu)^2}{2}$. You need to follow similar steps as above and using this conditional posterior result of $P(\sigma^2 \mid \mu, \mathbf{y})$ to integrate the full posterior with respect to σ^2.

To obtain analytical credible intervals, we can use these marginal posteriors.

Now that we are in multi-parameter setting, we can see that future observations also similar to parameter. Thus, the distribution of a future observation would be just another integration. $P(\tilde{\mathbf{y}} \mid \mathbf{y}) = \int\int P(\mid \mu, \sigma^2) P(\mu, \sigma^2 \mid \mathbf{y}) d\mu d\sigma^2$.

Semi-conjugate prior Now, let's consider this prior $P(\mu, \sigma^2) = P(\mu \mid \sigma^2)P(\sigma^2) = \text{Normal}(\mu_0, \sigma^2/\kappa_0)\text{Inv-}\chi^2(\nu_0, \sigma_0^2)$

Note that Inv-$\chi^2(\nu_0, \sigma_0^2)$ is a scaled-Inv-χ^2 distribution. Then $\frac{\sigma^2}{\nu_0\sigma_0^2} \sim$ Inv-$\chi^2(\nu_0)$. Thus $P(\sigma^2) \propto \left(\frac{\sigma^2}{\sigma_0^2\nu_0}\right)^{-\nu_0/2-1} \exp(-\frac{\nu_0\sigma_0^2}{2\sigma^2})$.

Result: The posterior will be $P(\mu, \sigma^2 \mid \mathbf{y}) = P(\mu \mid \sigma^2, \mathbf{y})P(\sigma^2 \mid \mathbf{y})$ where $P(\mu \mid \sigma^2, \mathbf{y}) \sim \text{Normal}(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}, \frac{\sigma^2}{n+\kappa_0})$ and $P(\sigma^2 \mid \mathbf{y}) = \text{Inv-chi}(n+\nu_0, (n-1)s^2 + \nu_0\sigma_0^2 + \frac{\kappa_0 n}{\kappa_0+n}(\bar{\mathbf{y}}-\mu_0)^2$.

Proof. I will show the posterior following some steps which are general enough to apply to any problem:

Step 1 Write down the full "proportional" posterior i.e. Likelihood $\times$ prior which is in this case:

$$\frac{1}{\sigma^n} \exp\left[-\frac{1}{2\sigma^2}\{(n-1)s^2 + n(\bar{\mathbf{y}}-\mu)^2\}\right] \times \frac{\sqrt{\kappa_0}}{\sigma} \exp\{-\frac{\kappa_0}{2\sigma^2}(\mu-\mu_0)^2\} \times \left(\frac{\sigma^2}{\sigma_0^2\nu_0}\right)^{-\nu_0/2-1} \exp(-\frac{\nu_0\sigma_0^2}{2\sigma^2}).$$

Step 2 Rearrange the terms. Bring together the common looking terms in terms of σ here (and ignoring few more constant multipliers from above like $\sqrt{\kappa_0}, (\sigma_0^2)^{\nu/2+1}$.

$$\frac{1}{\sigma^{n+1+\nu_0+2}} \exp\left[-\frac{1}{2\sigma^2}\{(n-1)s^2 + n(\bar{\mathbf{y}}-\mu)^2 + \kappa_0(\mu-\mu_0)^2 + \nu_0\sigma_0^2\}\right]$$

Step 3 Now we try to find posterior of μ. The result we use is that $P(\mu \mid \sigma^2, \mathbf{y}) \propto P(\mu, \sigma^2 \mid \mathbf{y})$ which is proportional to the above term in Step 2. Hence, to identify the posterior of μ, we collect terms involving μ in Step 2.

$$\exp\left[-\frac{1}{2\sigma^2}\{n(\bar{\mathbf{y}}-\mu)^2+\kappa_0(\mu-\mu_0)^2\}\right] \propto \exp\left[-\frac{1}{2\sigma^2}\{\mu^2(n+\kappa_0)-2\mu(n\bar{\mathbf{y}}+\kappa_0\mu_0)\}\right]\times\exp\left(-\frac{n\bar{\mathbf{y}}^2+\kappa_0\mu_0^2}{2\sigma^2}\right)$$

A bit of rearranging will give us (ignoring the last term as it does not involve μ), $\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\{\mu^2-2\mu\frac{(n\bar{\mathbf{y}}+\kappa_0\mu_0)}{(n+\kappa_0)}\right\}$. Now this can be re-written as $\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\{\mu^2-2\mu\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}+\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2-\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\}\right]$, which is,

$$\exp\left\{\frac{n+\kappa_0}{2\sigma^2}\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right\}\times\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\left(\mu-\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right].$$

Only the second term involves μ. Hence, the proportional conditional posterior for μ is $\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\left(\mu-\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right]$ which tells us the conditional distribution will be $\mathbf{Normal}(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}, \frac{\sigma^2}{n+\kappa_0})$.

Step 4 The terms we ignored along the way will be now added back (as they had terms involving σ^2), we rewrite step 2.

$$\frac{1}{\sigma^{n+\nu_0+2}}\exp\left[-\frac{1}{2\sigma^2}\{(n-1)s^2+\nu_0\sigma_0^2+n\bar{\mathbf{y}}^2+\kappa_0\mu_0^2\}\right]\exp\left\{\frac{n+\kappa_0}{2\sigma^2}\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right\}$$
$$\times\frac{1}{\sigma}\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\left(\mu-\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right]$$

Step 5 Using the derived conditional posterior for μ, we have $\int\frac{1}{\sigma}\exp\left[-\frac{n+\kappa_0}{2\sigma^2}\left(\mu-\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\right]d\mu = \frac{\sqrt{2\pi}}{\sqrt{n+\kappa_0}}$, (a constant). Hence the marginal posterior of σ^2 will proportional to,

$$P(\sigma^2 \mid \mathbf{y}) \propto (\sigma^2)^{-\frac{n+\nu_0}{2}-1}\exp\left[-\frac{1}{2\sigma^2}\{(n-1)s^2+\nu_0\sigma_0^2+n\bar{\mathbf{y}}^2+\kappa_0\mu^2-(n+\kappa_0)\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2\}\right].$$

We have $n\bar{\mathbf{y}}^2+\kappa_0\mu^2-(n+\kappa_0)\left(\frac{n\bar{\mathbf{y}}+\kappa_0\mu_0}{n+\kappa_0}\right)^2 = \frac{\kappa_0 n}{\kappa_0+n}(\bar{\mathbf{y}}-\mu_0)^2$, hence,

$$P(\sigma^2 \mid \mathbf{y}) \propto (\sigma^2)^{-\frac{n+\nu_0}{2}-1}\exp\left[-\frac{1}{2\sigma^2}\{(n-1)s^2+\nu_0\sigma_0^2+\frac{\kappa_0 n}{\kappa_0+n}(\bar{\mathbf{y}}-\mu_0)^2\}\right].$$

Following the above expression for Scaled-Inv-χ^2 distribution, we have $\sigma^2 \mid \mathbf{y}$ follows Scaled-Inv-χ^2 with parameters $\bar{\nu} = n+\nu_0$ and $\bar{\nu}\bar{\sigma^2} = (n-1)s^2+\nu_0\sigma_0^2+\frac{\kappa_0 n}{\kappa_0+n}(\bar{\mathbf{y}}-\mu_0)^2$.

□

Sampling: In Bayes, everything boils down to sampling. In a multi-parameter setting, we again use the method of composition for sampling.

Based on the Normal-inv-chisquared case we derived last time, we can consider the following sampling scheme. Through the following example, I also want to point out some tricks for sampling from transformation based ditribution.

```
 #Generate the data
sigmag <- 1
mug    <- 4
n      <- 1000
y <- rnorm(n, mug, sigmag)

#Prior hyperparameter
mu0    <- 0
kappa0 <- 0.1
nu0    <- 0.1
sigma0 <- 1

mubar        <- (kappa0*mu0 + n*mean(y))/(kappa0+n)
kappabar     <- kappa0 + n
nusig2bar    <- nu0*sigma0^2 + (n-1)*sd(y)^2 + ((kappa0*n)/(kappa0+n))*(mean(y)-mu0)^2
nubar        <- nu0+n

#Generate sigma

invchisam <- 1/rchisq(1000, nubar)
sigma2sam <- nusig2bar * invchisam #due to the relation between inv-chisq and scaled-invchi

musam     <- rnorm(1000, mubar, sqrt(sigma2sam/kappabar))

#Above code is same as
for(i in 1:1000)
{musam[i] <- rnorm(1, mubar, sqrt(sigma2sam[i]/kappabar))}

quantile(musam, probs = c(0.025, 0.975))
quantile(sigma2sam, probs = c(0.025, 0.975))

#Alternative way using LaplacesDemon package to generate 'scaled-chisq' directly and subsequently
#invert those samples to get scaled-invchisq.

sigma2samalt <- 1/LaplacesDemon::rinvchisq(1000, nubar, scale=nusig2bar/nubar)

musamalt     <- rnorm(1000, mubar, sqrt(sigma2samalt/kappabar))
```

```
#Above code is same as
for(i in 1:1000)
{musamalt[i] <- rnorm(1, mubar, sqrt(sigma2samalt[i]/kappabar))}

#predict a new observation
ytilde <- rnorm(1000, musam, sigmasam) #this produces one sample of predicted sample for each
                                       #posterior sample of mu and sigma.
#Above code is same as
for(i in 1:1000)
{ytilde[i] <- rnorm(1, musam[i], sigmasam[i])}

quantile(musamalt, probs = c(0.025, 0.975))
quantile(sigma2samalt, probs = c(0.025, 0.975))
```

In the above sampling step, to sample from Inv-chisq, we first sampled from chisq and then just invert those samples to get samples of inv-chisq. Also, to generate scaled inv-chisq samples, we can again use the relation between scaled inv-chisq and inv-chisq.

Regression Regression is one of the most important techniques in a statistician's toolbox. In any real data setting, we always try to identify a relation between a response and a bunch of predictors. The most common regression model is the simple linear regression model with Gaussian error, as the response is often continuous. Having a continuous response simplifies our inference problem greatly. The priors are exactly similar to the Gaussian likelihood case with unknown mean and variance.

The overall structure of the model heavily depends on the data type at hand. Specifically, our assumption for the 'error model' will lead to different types of linear regression models. For generality, these models are called generalized regression model.

Linear regression with non-informative prior:

The likelihood $\propto \frac{1}{\sigma^n}\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\}/(2\sigma^2)\}$, where $\mathbf{y} = (y_1, \ldots, y_n)$. And the prior: $(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}$.

Hence the proportional posterior is $P(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}) \propto \frac{1}{\sigma^{n+2}}\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\}/(2\sigma^2))$

The part that involves $\boldsymbol{\beta}$ is $\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}/(2\sigma^2)\} = \exp\{-(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}+\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}+\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})/(2\sigma^2)\}$

Some results from linear regression: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. The error $\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y}-\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = (\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y}$ and $(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{X} = \mathbf{X} - \mathbf{X} = 0$. Further $(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{y}^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y} = \mathbf{y}^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y}$ as $(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T) = (\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)$ (verify!)

We can write $(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}+\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}+\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta}) = \{(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})+(\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})\}^T\{(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})+(\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})\} = (\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T((\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})+(\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta})+2(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{X}\hat{\boldsymbol{\beta}}-\mathbf{X}\boldsymbol{\beta}) =$

$(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T((\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + 2\mathbf{y}^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$. (Verify)

Since, $(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{X} = 0$, we have $(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T((\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + 2\mathbf{y}^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T((\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = 0$.

Hence, $\frac{1}{\sigma^{n+2}}\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\}/(2\sigma^2)) = \frac{1}{\sigma^{n-d-1+2}}\frac{1}{\sigma^{d+1}}\exp\{-(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/(2\sigma^2)\} = \frac{1}{\sigma^{n-d-1+2}}\exp\{-(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2)\}\frac{1}{\sigma^{d+1}}\exp\{-(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/(2\sigma^2)\}$.

The part $\frac{1}{\sigma^{d+1}}\exp\{-(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})/(2\sigma^2)\}$ contributes to the conditional normal distribution part of $\boldsymbol{\beta}$ where we get $\boldsymbol{\beta} \mid \sigma^2, \mathbf{y} \sim \text{Normal}(\hat{\boldsymbol{\beta}}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$. Hence, $\int \frac{1}{\sigma^{d+1}}\exp\{-(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/(2\sigma^2)\}d\boldsymbol{\beta} \propto C$, some constant.

We have,

$$
\begin{aligned}
P(\sigma^2 \mid \mathbf{y}) &= \int P(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y})d\boldsymbol{\beta} \\
&\propto \int \frac{1}{\sigma^{n-d-1+2}}\exp\{-(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2)\}\frac{1}{\sigma^{d+1}}\exp\{-(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/(2\sigma^2)\}d\boldsymbol{\beta} \\
&= \frac{1}{\sigma^{n-d-1+2}}\exp\{-(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2)\}\int \frac{1}{\sigma^{d+1}}\exp\{-(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\mathbf{X}^T\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/(2\sigma^2)\}d\boldsymbol{\beta} \\
&= C\frac{1}{\sigma^{n-d-1+2}}\exp\{-(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2)\} \\
&= C(\sigma^2)^{-(n-d-1)/2-1}\exp\{-(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/(2\sigma^2)\}
\end{aligned}
$$

This leads to an Inv-χ^2-distribution for σ^2 with parameters $(n-d-1)/2$ and $(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = (n-d-1)s^2$, where $s^2 = \frac{(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})}{n-d-1} = \frac{\mathbf{y}^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y}}{n-d-1}$.

Hence, $P(\sigma^2 \mid \mathbf{y}) \propto C(\sigma^2)^{-(n-d-1)/2-1}\exp\{-(n-d-1)s^2/(2\sigma^2)\}$ and thus, $\sigma^2 \mid \mathbf{y} \sim \text{Inv-}\chi^2(n-d-1, s^2)$.

$P(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}) = P(\boldsymbol{\beta} \mid \sigma^2, \mathbf{y})P(\sigma^2 \mid \mathbf{y}) = \text{Normal}(\hat{\boldsymbol{\beta}}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})\text{Inv-}\chi^2(n-d-1, s^2)$

Part 5

Linear regression The conjugate prior for linear regression with Gaussian noise i.e. $y_i \sim \text{Normal}(\mathbf{x}_i\boldsymbol{\beta}, \sigma^2)$ is $\boldsymbol{\beta} \sim \text{Normal}(\beta_0, \sigma^2\boldsymbol{\Sigma}_0)$ with σ^2 follows scaled Inv-chisquared or inverse gamma. In case of NIG prior, we set σ^2 follow Inv-Gamma(a_0, b_0). The overall prior is then $P(\boldsymbol{\beta}, \sigma^2) = P(\boldsymbol{\beta} \mid \sigma^2)P(\sigma 2) = \text{Normal}(\beta_0, \sigma^2\boldsymbol{\Sigma}_0) \times \text{Inv-Gamma}(a_0, b_0)$. Then the joint posterior can be evaluated in the form of Method of Composition type structure: $P(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}) = P(\boldsymbol{\beta} \mid$

$\sigma^2, \mathbf{y})P(\sigma^2 \mid \mathbf{y})$. Under the above priors, the posteriors are:

$$P(\sigma^2 \mid \mathbf{y}) \sim \text{Inv-Gamma}(\bar{a}, \bar{b}),$$
$$P(\boldsymbol{\beta} \mid \sigma^2, \mathbf{y}) \sim \text{Normal}(\bar{\boldsymbol{\beta}}, \sigma^2 \bar{\boldsymbol{\Sigma}}),$$

where,

$$\begin{aligned}
\bar{\boldsymbol{\Sigma}} &= (\boldsymbol{\Sigma}_0^{-1} + \mathbf{X}^T\mathbf{X})^{-1}, \\
\bar{\boldsymbol{\beta}} &= \bar{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0 + \mathbf{X}^T\mathbf{y}), \\
\bar{a} &= a_0 + n/2, \\
\bar{b} &= b_0 + \boldsymbol{\beta}_0^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0/2 + \mathbf{y}^T\mathbf{y}/2 - \bar{\boldsymbol{\beta}}^T\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{\beta}}/2.
\end{aligned}$$

Weakly informative version of the above prior can be set by letting $a_0 = b_0 = 0.1$ (very small) and $\boldsymbol{\Sigma}_0 = 100\mathbf{I}$ i.e. a diagonal matrix with 100 (in this case or even something larger) in the diagonal with $\boldsymbol{\beta}_0 = 0$, a zero vector.

Zellner g-prior is a particular case of NIG prior where we set $\boldsymbol{\Sigma}_0 = g(\mathbf{X}^T\mathbf{X})^{-1}$ for some constant positive constant g. A weakly informative version of this prior is obtained by setting, $g = 100$ along with all the above-mentioned choices for a_0, b_0 and $\boldsymbol{\beta}_0$.

```
set.seed(1)
n1 = 1000
# Generate data
x1 <- rnorm(n1)
z1 <- rnorm(n1)
#z1 <- (diag(n1) - tcrossprod(x1)/sum(x1^2)) %*% z1
e <- rnorm(n1)
y1 <- 2*x1+e

n=30

x <- x1[1:n]
z <- z1[1:n]
y <- y1[1:n]

X <- cbind(x,z)

posteriorsam <- 5000

#Hyper-parameters
a <- 0.1 # For sigma
b <- 0.1 # For sigma
betas2prior <- 100 # Prior variance V_beta= betas2prior*diag(2)
```

```
betamuprior <- rep(0,2)# mu_beta

beta.Ivar <- crossprod(X) + diag(1/betas2prior,2,2) #inverse of V*
beta.var <- solve(crossprod(X) + diag(1/betas2prior,2,2)) #This is my V*
beta.hat <- beta.var %*% (diag(1/100,2,2)%*%betamuprior + crossprod(X, y)) #This is mu*

#posterior samples of the variance
a1p <- a+n/2 #Parameter for marginal posterior of sigma, this a*
b1p <- b+t(betamuprior)%*%diag(1/100,2,2)%*%betamuprior/2+crossprod(y)/2
-t(beta.hat)%*%beta.Ivar%*%beta.hat/2 #This is b*

sigma2s <- 1/rgamma(posteriorsam, a1p, b1p) #Sample first from gamma, then invert it
#to get a sample from inverse-gamma

library(mvtnorm)

#posterior samples of the regression coefficients for each sample of sigma
betas <- lapply(sigma2s, FUN = function(x){rmvnorm(1,mean=beta.hat,sigma=x*beta.var)})
betas <- t(matrix(unlist(betas), 2, posteriorsam))

py1 <- 0
C <- 4 # constant to avoid singularity
py1s <- lapply(1:posteriorsam,
FUN = function(i){prod(C*dnorm(y,mean = as.vector(X%*%betas[i,]), sd = sqrt(sigma2s[i])))})
py1s <- unlist(py1s)

py1hat <- 1/(mean(1/py1s)) #Harmonic mean of the likelihood as the importance
#sampling estimate of the p(y|H_a)

quantile(betas[,1], probs = c(0.025, 0.975)) #This is will give you credible intervals
quantile(betas[,2], probs = c(0.025, 0.975))
```

Part MCMC

1 Basics

When we have multiple parameters, we use method of composition (MoC) for sampling. To apply MoC, we need to be able to get marginal posterior for one of the parameters. It might become very hard. We have seen that for semi-conjugate priors, it is hard to apply MoC for this reason.

Thus, to tackle such problems, Markov chain Monte Carlo is developed. 'Monte Carlo' is a fancy name for sampling. 'Markov chain' is the new part. In general, if any sequence of events

have lag-1 dependence, it is called a Markov chain. The distribution of x[i+1] depended on x[i]. It is called lag-1 due to that 1. It would have been lag-2 dependence if x[i+1] depended on x[i] as well as x[i-1] or x[i-1] alone. Like daily temperature usually exhibit such dependence. The temperature of Tuesday is expected to depend on that of Monday. Then it will be lag-1. However, if temperature of Tuesday is expected to depend on that of Monday and Sunday. Then it will be lag-2. There may be higher order dependence too, meaning present observation may depend on longer history.

More generally, such interdependent sequence of events is called a Markov chain. the value transit to the other based on some transition probability which is the key part.

MCMC requires lesser mathematical derivations than MoC and thus is more popular. However, lesser math comes at a price. You need to perform a lot of post-hoc analysis to ensure 'correctness' of the samples.

Autocorrelation: Correlation computes the association between two different variables like age and height or height and weight. Autocorrelation computes the association within itself. It is in general applied to the datasets that are collected sequentially, like daily temperature data, stock market data, etc. In temperature data, for example, we may be interested to know 'What the association between the temperatures on the two consecutive days?' Say we have daily temperature data as 50,53,52,49,48,49,53,54,51. How do you answer the above Qn?

Autocorrelations are defined in terms of lag-'number'. Lag-1 autocorrelation is the correlation $\gamma(1)$ =cov(X_t, X_{t-1}). Similarly, Lag-h autocorrelation is the correlation $\gamma(h)$ =cov(X_t, X_{t-h}). In R, `acf()` computes these correlations and also provide results on their significance (will see).

In an ideal posterior sampling, all the samples are expected to be independent. Like for conjugate prior, we drew 5000 samples of σ simultaneously. Hence, these samples are all independent. This means that $\gamma(h) = 0$ or statistically insignificant for any $h > 0$.

However, by construction, MCMC samples are not fully independent samples of the posterior. There is some dependence. We see that below example.

1.1 Gibbs sampling:

Gibbs sampling requires some mathematical derivations, but much less than MoC. Semi-conjugate priors for linear regression: Linear regression model: $y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i$, where $\epsilon_i \sim$ Normal$(0, \sigma^2)$ with $i = 1, \ldots, n$. Priors:$\boldsymbol{\beta} \sim$ Normal$(0, 100\mathbf{I})$ and $\sigma^2 \sim$ Inv-Ga$(0.1, 0.1)$. Here $\mathbf{I}$ is the identity matrix.

How to derive conditional posterior: Let us assume there K many parameters $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$ and we use the notation $\boldsymbol{\theta}_{-j}$ to denote all the entries in $\boldsymbol{\theta}$, removing θ_j i.e. $\boldsymbol{\theta}_{-j} = \{\theta_1, \ldots, \theta_{-1}, \theta_{j+1}\theta_K\}$. Note that $P(\theta_j \mid \boldsymbol{\theta}_{-j}, \mathbf{y}) \propto P(\boldsymbol{\theta} \mid \mathbf{y}) \propto P(\mathbf{y} \mid \boldsymbol{\theta})P(\boldsymbol{\theta})$, which is Likelihood×prior, the proportional posterior.

- We first write the complete proportional posterior.
- Then collect all the relevant terms.
- Simplify those relevant terms to check if it leads to any known distribution class.

For our above-mentioned Bayesian problem with sample size n:

- Proportional posterior: $\frac{1}{\sigma^n}\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/(2\sigma^2)\}\exp(-\boldsymbol{\beta}^T\boldsymbol{\beta}/(2\cdot 100))(\sigma^{-2})^{0.1-1}\exp(-\frac{0.1}{\sigma^2})$.
- For conditional posterior of σ: Collecting the relevant terms which are $\frac{1}{\sigma^n}\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/(2\sigma^2)\}(\sigma^{-2})^{0.1-1}\exp(-\frac{0.1}{\sigma^2}) = (\sigma^{-2})^{n/2+0.1-1}\exp\left\{-\frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/2+0.1}{\sigma^2}\right\}$ which leads to Inv-Ga$(n/2+0.1, (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/2+0.1)$.
- Similarly, for $\boldsymbol{\beta}$: Collecting the relevant terms which are $\exp\{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/(2\sigma^2)\}\exp(-\boldsymbol{\beta}^T\boldsymbol{\beta}/(2\cdot 100)) \propto \exp\left[\frac{1}{2}\{2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y}-\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X})\boldsymbol{\beta}-\boldsymbol{\beta}(\frac{1}{100}\mathbf{I})\boldsymbol{\beta}\}\right] = \exp\left[\frac{1}{2}\{2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y}-\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X}+\frac{1}{100}\mathbf{I})\boldsymbol{\beta}\}\right]$.
- If $\boldsymbol{\beta} \sim$ MVN$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then it's density is $\frac{1}{\sqrt{2\pi}\det(\boldsymbol{\Sigma})^{1/2}}\exp\{-(\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})\} = \frac{1}{\sqrt{2\pi}\det(\boldsymbol{\Sigma})^{1/2}}\exp\{-\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}+2\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\}$. Now part of this density that involves $\boldsymbol{\beta}$ is $\exp\{-\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}+2\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\}$. Comparing this part with above derived proportional full conditional, we must have $\boldsymbol{\Sigma}^{-1} = \mathbf{X}^T\mathbf{X}+\frac{1}{100}\mathbf{I} \implies \boldsymbol{\Sigma} = (\mathbf{X}^T\mathbf{X}+\frac{1}{100}\mathbf{I})^{-1}$ and $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \mathbf{X}^T\mathbf{y} \implies \boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{X}^T\mathbf{y}$.

These conditional distributions will be used for sampling.

```
#data generation
set.seed(1)
p <- 1
X <- matrix(rnorm(100*p), 100, p)

beta0 <- rnorm(10, sd = 3)

#beta0[which(abs(beta0) < 3)] <- 0

y <- rnorm(100, X %*% beta0, 1)


#Posterior sampling via Gibbs
n <- nrow(X)

library(mvtnorm)

#Hyper-parameters
a1 <- 0.1
b1 <- 0.1
```

```
betas2prior <- 100
betamuprior <- rep(0,p)

###Initialization

beta <- rep(0, p)
sigma <- 0.1

itr <- 0

#Define lists to store postburn samples
sigmals <- list()
betals  <- list()

Total_itr <- 1000
burn      <- 500 #Sample before itr==burn are in burn-in period.

while(itr < Total_itr){
  #start_time <- Sys.time()

  itr <- itr + 1
  ####Sample sigma from its full conditional given a sample of beta###############
  #This is the posterior parameters for (sigma|beta)
  a1p <- a1+n/2
  b1p <- b1+crossprod(y-X%*%beta)/2

  sigma <- sqrt(1/rgamma(1, a1p, b1p))

  ####Sample beta from its full conditional given a sample of sigma##########
  beta.Ivar <- crossprod(X)/sigma^2 + diag(1/betas2prior,p,p)

  beta.var <- solve(crossprod(X)/sigma^2 + diag(1/betas2prior,p,p))
  beta.hat <- beta.var %*% (crossprod(X, y)/sigma^2+diag(1/betas2prior,p,p)%*%betamuprior)

  beta <- array(rmvnorm(1,mean=beta.hat,sigma=beta.var))

  ####Store the posterior samples
  if(itr > burn){
    sigmals[[itr - burn]] <- sigma
    betals[[itr - burn]]  <- beta
  }

  #end_time <- Sys.time()
}
```

```
Reduce('+', sigmals)/500
```

Prediction with for new X.

```
Xnew <- matrix(rnorm(1*p), 1, p) #Generate new X
ypred <- rep(0, 500) #store the the generated predictions for each MCMC sample
for(itr in 1:500){
    ypred[itr] <- rnorm(1, Xnew %*% betals[[itr]], sigmals[[itr]])
}
mean(ypred)
quantile(ypred, probs = c(0.0025, 0.975))
```

Above code works for any p. In the above example, we are moving in the following manner: $(\sigma^0, \boldsymbol{\beta}^0) \to (\sigma^1 \to \boldsymbol{\beta}^1) \to (\sigma^2 \to \boldsymbol{\beta}^2) \cdots$

Finally, $\{\sigma^{501}, \ldots, \sigma^{1000}\}$ are our posterior samples of σ after confirming that these are 'good' samples. Similarly for $\boldsymbol{\beta}$.

What we see different above?:

- The samples are generated by conditioning on the rest of the parameters.
- Thus, the samples become dependent on the previously generated samples. (Clue: we are using a while loop for sampling, which can not parallelized.)

Potential pitfall:

- Will my estimate depend on the starting value?
- For Bayesian inference, we need the posterior samples to be independent. However, these samples are dependent.
- The sampler can not be parallelized (like for MoC, we could use `apply` while sampling $\boldsymbol{\beta}$ for each σ and also we generated all the samples of σ together as used the marginal posterior for σ, not conditional).

Elaboration of first point: The estimate may depend on initialization if the posterior likelihood is multi-modal. It might get trapped in a local mode. Thus, it is advisable to run the MCMC chains for multiple starting points. This is also fairly common in any optimization routine.

Elaboration of second point: We start with some $\boldsymbol{\beta}^0$ and σ^0. Given $\boldsymbol{\beta}^0$, we sample/update σ and move from $\sigma^0 \to \sigma^1$. This σ^1 is the sample `sqrt(1/rgamma(1, a1p, b1p))`. Next, given this new σ^1, we draw a sample of $\boldsymbol{\beta}$ and move from $\boldsymbol{\beta}^0 \to \boldsymbol{\beta}^1$. This $\boldsymbol{\beta}^1$ is the sample `array(rmvnorm(1,mean=beta.hat,sigma=beta.var))`

In the next iteration: We generate σ given $\boldsymbol{\beta}^1$ and further move from $\sigma^1 \to \sigma^2$.

If we generalize above step for K-many parameters, we simply update each of the K-parameters one by one given all the other. For example, our $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{10})$. We can then sample β_1 from $P(\beta_1 \mid \beta_2, \ldots, \beta_{10}, \sigma)$, β_2 for $P(\beta_2 \mid \beta_1, \beta_3, \ldots, \beta_{10}, \sigma)$, ….

However, we are sampling all the component of $\boldsymbol{\beta}$ together. We can also sample $\beta_{1:5}$ from $P(\beta_{1:5} \mid \beta_{6:10}, \sigma)$ and $\beta_{6:10}$ from $P(\beta_{6:10} \mid \beta_{1:5}, \sigma)$. This is called *Blocked Gibbs.*

Elaboration of third point: Why we need to do it sequentially but not parallelly like in the past? The answer to that is in the probability. We have $P(\boldsymbol{\beta}, \sigma \mid \mathbf{y}) = P(\boldsymbol{\beta} \mid \sigma, \mathbf{y})P(\sigma \mid \mathbf{y})$. But we are generating sample from conditional posteriors for both of the two parameters i.e. we are sampling $\boldsymbol{\beta}$ from $P(\boldsymbol{\beta} \mid \sigma, \mathbf{y})$ and sampling σ from $P(\sigma \mid \boldsymbol{\beta}, \mathbf{y})$, AND $P(\boldsymbol{\beta}, \sigma \mid \mathbf{y}) \neq P(\boldsymbol{\beta} \mid \sigma, \mathbf{y})P(\sigma \mid \boldsymbol{\beta}, \mathbf{y})$.

This has some commonality with standard multiparameter optimization routines. But they are conceptually not the same. In any optimization routine, we update the parameters sequentially in a loop. MCMC can be considered as a sampling-based alternative of the optimization. However, MCMC requires much larger number of iterations to converge.

In the above problem, we could explicitly obtain express the conditional posterior in terms of known distribution class. $P(\sigma \mid \boldsymbol{\beta}, \mathbf{y})$ is inverse-gamma and $P(\boldsymbol{\beta} \mid \sigma, \mathbf{y})$ is normal. The above prior is thus called semi-conjugate. However, life may not always be that simple i.e. semi conjugate may not always be appropriate.

1.2 General Metropolis-Hastings (MH)

It may not be possible to arrive at a known distribution class for all the full conditional posteriors. For such situations, we consider the MH algorithm for sampling.

How does it work?

- Let θ^t be the value of θ at t-th MCMC iteration.
- For $t+1$-th iteration, we propose a new value of θ_c (which is called candidate).
- Next would be to decide whether this new value should be considered as a new sample/update for θ. If it gets selected, then we set $\theta^{t+1} = \theta_c$ otherwise set $\theta^{t+1} = \theta^t$.

How do we propose this new value and decide whether to accept it or not?

- Transition: Choose a probability distribution with one of the parameters being θ^t and the expectation under this distribution should ideally be θ^t. Example: If we choose Normal(θ^t, ϵ^2). The expectation of this distribution is θ^t. Or we may also sample from Ga(a, a) for any a and multiply that with θ^t i.e. if $y = \theta^t x$, where $x \sim$ Ga(a, a), we

have $\mathbb{E}(y) = \theta^t$ as $\mathbb{E}(x) = 1$. This distribution is called *Transition probability* and denoted by q. The probability of transitioning from θ^t to θ_c is denoted by $q(\theta_c \mid \theta^t)$ when $q(\cdot \mid \theta^t) = \text{Normal}(\theta^t, \epsilon^2)$, we can calculate $q(\theta_c \mid \theta^t)$=`dnorm`$(\theta_c \mid \theta^t, 0, \epsilon)$

- Acceptance: We are not accepting all the updates at each move. It is a probabilistic move. It relies on the acceptance probability, which is computed as $p = \frac{P(\mathbf{y}|\theta_c)P(\theta_c)}{P(\mathbf{y}|\theta^t)P(\theta^t)} \frac{q(\theta^t|\theta_c)}{q(\theta_c|\theta^t)} = \frac{P(\theta_c|\mathbf{y})}{P(\theta^t|\mathbf{y})} \frac{q(\theta^t|\theta_c)}{q(\theta_c|\theta^t)}$, where θ_c is the candidate-update of θ.

1.3 Random walk MH:

MH sampling requires no mathematical derivations. You just need to be able to write down the likelihood. MH sampling can be applied for ANY Likelihood and ANY choice of prior. Well, it comes at a price. We study that later.

Random walk MH sets the transition probability to Normal distribution. Hence, for random-walk MH, the q is always Normal.

```
#Data generation

set.seed(1)
X <- matrix(rnorm(100), 100, 1)

beta0 <- rnorm(1, sd = 3)

y <- rnorm(100, X %*% beta0, 1)


#Consider normal prior for beta
priormean <- 0
priorvar  <- 100

#Consider inverse-gamma prior for sigma
al0 <- 0.1
be0 <- 0.1


chainseed <- function(seed){
  set.seed(seed)
#Initialization
beta <- 1 #lm(y~X-1)$coefficients
sigma <- 1 #sum((lm(y~X-1)$residuals)^2)/(100-2)
llhood <- sum(dnorm(y, X%*%beta, sigma, log = T)) #log-likelihood of the data
prhood <- dnorm(beta, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigma^2, al0, be0) #prior log
```

```
itr <- 0

#Define lists to store postburn samples
betals <- sigmals <- list()

Total_itr <- 10000
burn      <- 5000

epsilon1 <- 1e-1
flag1    <- 0

epsilon2 <- 1e-1
flag2    <- 0

while(itr < Total_itr){
  itr <- itr + 1

  betac <- beta + rnorm(1, sd = epsilon1)
#If epsilon1 is small, then betac and lambda are very close => acceptance prob close to 1
#=> force to increase acceptance

#If epsilon1 is large, then betac and lambda will be far
#=> acceptance prob will likely to reduce => force to decrease acceptance

llhoodc <- sum(dnorm(y, X%*%betac, sigma, log = T)) #log-likelihood of the data
prhoodc <- dnorm(betac, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigma^2, al0, be0) #prior l

#llhoodc + prhoodc is actually posterior log-likelihood for betac, sigma

D <- llhoodc + prhoodc - (llhood + prhood)


u <- runif(1)

#posterior likelihood at candidate > posterior likelihood at current value
#=> exp(D) > 1 => D > 0

#When you generate u as runif(1), 'u' bounded [0,1]. log(u) < 0 for all possibilities of u.
#log(u) < D for any 'u', generated.

#D is small => exp(D) is small
#=> posterior likelihood at candidate < posterior likelihood at current value
#The condition log(u) < D will hold for smaller possibilities of u.
```

```
if(log(u) < D){ #u < exp(D) [=ratio of likelihoods] P(log(u)<D)=P(u<exp(D))=exp(D)
  beta <- betac
  llhood <- llhoodc
  prhood <- prhoodc

  flag1 <- flag1 + 1 #To count number of accepts
}

betals[[itr]] <- beta #storing the posterior samples

sigmac <- exp(log(sigma) + rnorm(1, sd = epsilon2))

llhoodc <- sum(dnorm(y, X%*%beta, sigmac, log = T)) #log-likelihood of the data
prhoodc <- dnorm(beta, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigmac^2, al0, be0) #prior l

#llhoodc + prhoodc is actually posterior log-likelihood for betac, sigma

D <- llhoodc + prhoodc - (llhood + prhood)


u <- runif(1)

#posterior likelihood at candidate > posterior likelihood at current value
#=> exp(D) > 1 => D > 0

#When you generate u as runif(1), 'u' bounded [0,1]. log(u) < 0 for all possibilities of u.
#log(u) < D for any 'u', generated.

#D is small => exp(D) is small
#=> posterior likelihood at candidate < posterior likelihood at current value
#The condition log(u) < D will hold for smaller possibilities of u.

if(log(u) < D){ #u < exp(D) [=ratio of likelihoods] P(log(u)<D)=P(u<exp(D))=exp(D)
  sigma <- sigmac
  llhood <- llhoodc
  prhood <- prhoodc

  flag2 <- flag2 + 1 #To count number of accepts
}

sigmals[[itr]] <- sigma #storing the posterior samples

#For M-H sampler + univariate case, usually 0.45 is an optimal acceptance rate.
```

```
if(itr %% 100==0){ #Auto-tuning step
  ar <- flag1 / itr #Acceptance rate
  if(ar < 0.3){ #Acceptance is too low
    epsilon1 <- epsilon1 / 2
  }

  if(ar > 0.5){ #Acceptance is too high
    epsilon1 <- epsilon1 * 2
  }
  print(ar) #To track the acceptance rate

  ar <- flag2 / itr #Acceptance rate
  if(ar < 0.3){ #Acceptance is too low
    epsilon2 <- epsilon2 / 2
  }

  if(ar > 0.5){ #Acceptance is too high
    epsilon2 <- epsilon2 * 2
  }
  print(ar) #To track the acceptance rate
}
}
  return(list(betap=betals,sigmap=sigmals))
}

Total_itr <- 10000
burn      <- 5000

samplesout <- chainseed(1)
```

Convergence diagnostics

Things to check:

1. Autocorrelation of the chain.
2. Thin the generated samples to reduce the autocorrelation.
3. Effective sample size of the chain.
4. Whether the performance of the sampler depend on the starting value. (It is often good to devise a strategy for initialization of the parameters and make that a default.)

```
betasamples  <- samplesout$betap[(burn+1):Total_itr]
sigmasamples <- samplesout$sigmap[(burn+1):Total_itr]
```

```
betas <- unlist(betasamples)
acf(betas) #Checking mixing by autocorrelation
acfvals <- acf(betas)

#Effective sample size. This number below essentially tells us our set of generated values
#correspond to how many independent samples.

M <- 7 #choice of M depends on acfvals$acf. abs(acfvals$acf) should always decrease.
#Then we choose M such that abs(acfvals$acf) is small enough (below th blue line in the plot)
#Here we choose 7 looking at the plot.

5000/(1+2*sum(acfvals$acf[1:M]))

#Depending on above ACF plot collect each 7-th sample

index <- (1:(5000/7))*7 - 6 #This is called thinning to reduce correlation

acf(betas[index]) #Re-checking mixing by autocorrelation

mean(betas[index]) #posterior mean
plot(density(betas[index])) #posterior density

quantile(betas[index], probs = c(0.025, 0.975)) #Equal tail 95% CI

library(coda)
chain <- mcmc(betas)

summary(chain)
plot(chain)

samplesout2 <- chainseed(2)

betasamples2  <- samplesout2$betap[(burn+1):Total_itr]
sigmasamples <- samplesout2$sigmap[(burn+1):Total_itr]

betasout2 <- unlist(betasamples2)

betas2 <- unlist(betasout2[(burn+1):Total_itr])
chain2 <- mcmc(betas2)

combinedchains = mcmc.list(chain, chain2)
plot(combinedchains)
gelman.diag(combinedchains)
gelman.plot(combinedchains)
```

```
library(BayesianTools)
correlationPlot(data.frame(chain))
```

Points to note

- Whereas in case of Gibbs, we were accepting all the updates. **Prove this mathematically that the acceptance probability of the Gibbs sampling is 1.**
- Random walk MH is a special type of MH. This the most used one as for this case, we have $P(\lambda_c \mid \lambda) = P(\lambda \mid \lambda_c)$.
- However, random walk updating requires the parameter to unrestricted which is why I am updating the log(parameter).
- The optimal acceptance rate is around 0.35-0.45 for random walk MH and it decreases with the dimension of the parameter. Usually keeping it between 0.3-0.5 produces good results.
- In random-walk MH, the acceptance is solely governed by the difference in log-likelihood and thus making it increasingly similar to optimization routines.

How to compute $q(\lambda_c \mid \lambda)$ and $q(\lambda \mid \lambda_c)$ for non-random walk situation. First, check why they are equal for random walk? The updating formula, we are considering, is `log(lambda_c)=log(lambda)+up` with up$\sim$Normal(0, epsilon^2). Hence, to go back to lambda from lambda_c, we require up_1 such that log(lambda)=log(lambda_c)+up_1 with up_1$\sim$Normal(0, epsilon^2). As $q(\lambda_c \mid \lambda) = P($up =log(lambda_c)-log(lambda))=`dnorm(log(lambda_c)-log(lambda), 0, epsilon)`

and $q(\lambda \mid \lambda_c) = P($up =log(lambda)-log(lambda_c))=`dnorm(log(lambda)-log(lambda_c), 0, epsilon)`.

For normal distribution,

`dnorm(log(lambda_c)-log(lambda), 0, epsilon)=dnorm(log(lambda)-log(lambda_c), 0, epsilon)`.

Now, if our updating equation is `lambda_c=q*lambda`, where m$\sim$Gamma(s, s). What are $q(\lambda_c \mid \lambda)$ and $q(\lambda \mid \lambda_c)$? Note that this is not random walk MH, hence these two are not equal.

Table 1: Transformation for making a restricted parameter unrestricted

Given parameter	Transformation	Inverse transformation
If $\theta > 0$	Take log transformation and set $\theta_1 = \log(\theta)$	$\theta = \exp(\theta_1)$
Bounded: If $\theta \in (a, b)$	Set $\lambda_1 = \log(\theta'/(1-\theta'))$ where $\theta' = \frac{\theta - a}{b-a}$	$\theta = (b-a)\frac{\exp(\lambda_1)}{1+\exp(\lambda_1)} + a$ (Verify!)

1.4 Gradient based MH:

This MH sampling method again requires no mathematical derivations on the distributions. You just need to be able to write down the likelihood and its derivatives. It is increasingly similar to optimization methods.

Tips for applying gradient based MH:

- Since gradient of the log-likelihood can be both positive or negative, it is advisable to make the MH move for the unrestricted parameter irrespective of the prior. Example: in the following problem, you may consider gamma prior for λ, but still make the MH move on $\log(\lambda)$ since $\log(\lambda)$ is unrestricted.

- The optimal acceptance rate is slightly higher than random walk MH. Usually between 0.45-0.7 produce good results.

The candidate update will, in this case, depend on the gradient of the posterior log-likelihood. Specifically $\theta_c = \theta^t + \frac{\epsilon}{2}\frac{\partial \log P(\theta|\mathbf{y})}{\partial \theta}|_{\theta=\theta^t} + \delta$, where δNormal$(0, \epsilon)$. Again we may want to apply above step on a transformation of the original parameter as the candidate may be both positive or negative on application of above equation.

This updating step is similar to Gradient ascent, where we try to maximize the objective function.

The proportional posterior likelihood of the following problem is $e^{-n\lambda}\lambda^{\sum_{i=1}^n y_i} \exp\{-(\log(\lambda)-\mu)^2/(2\sigma^2)\}$ assuming that the prior is $\log(\lambda) \sim$ Normal(μ_0, σ_0^2). Then in log-scale it will be $n\lambda + \log(\lambda)\sum_{i=1}^n y_i - (\log(\lambda)-\mu_0)^2/(2\sigma^2)$ If we write the above posterior log-likelihood in terms of $\lambda_1 = \log(\lambda)$, we get $-n\exp(\lambda_1) + \lambda_1 \sum_{i=1}^n y_i - (\lambda_1 - \mu_0)^2/(2\sigma^2)$ Then the derivative of the above with respect to λ_1 is $-n\exp(\lambda_1) + \sum_{i=1}^n y_i - (\lambda_1 - \mu_0)/(\sigma_0^2)$

```
#Data generation

set.seed(1)
X <- matrix(rnorm(100), 100, 1)

beta0 <- rnorm(1, sd = 3)

y <- rnorm(100, X %*% beta0, 1)


#Consider normal prior for beta
priormean <- 0
priorvar  <- 100

#Consider inverse-gamma prior for sigma
al0 <- 0.1
be0 <- 0.1


chainseed <- function(seed){
  set.seed(seed)
  #Initialization
  beta <- 1 #lm(y~X-1)$coefficients
  sigma <- 1 #sum((lm(y~X-1)$residuals)^2)/(100-2)
  llhood <- sum(dnorm(y, X%*%beta, sigma, log = T)) #log-likelihood of the data
```

```
  prhood <- dnorm(beta, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigma^2, al0, be0) #prior l

  itr <- 0

  n <- nrow(X)
  p <- ncol(X)

  #Define lists to store postburn samples
  betals <- sigmals <- list()

  Total_itr <- 10000
  burn      <- 5000

  epsilon1 <- 1e-5
  flag1    <- 0

  epsilon2 <- 1e-2
  flag2    <- 0

  derivbeta <- function(x){

    ret <- + t(X) %*% (y - X %*% x) / sigma^2 - (x-priormean) / priorvar

    return(ret)
  }

  derivsig <- function(x){

    ret <- -n + crossprod(y - X %*% beta)/(exp(x))^2

    ret <- ret - 2*(al0-1) + 2*(be0-1)/(exp(x))^2

    return(ret)
  }

  while(itr < Total_itr){
    itr <- itr + 1

    betac <- beta + derivbeta(beta)*epsilon1/2 + rnorm(1, sd = epsilon1)
#If epsilon1 is small, then betac and lambda are very close => acceptance prob close to 1
#=> force to increase acceptance

#If epsilon1 is large, then betac and lambda will be far
#=> acceptance prob will likely to reduce => force to decrease acceptance
```

```
llhoodc <- sum(dnorm(y, X%*%betac, sigma, log = T)) #log-likelihood of the data
prhoodc <- dnorm(betac, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigma^2, al0, be0) #prior l

#llhoodc + prhoodc is actually posterior log-likelihood for betac, sigma

#log{q(current value|candidate)}
q1 <- dnorm(beta-derivbeta(betac) * epsilon1/2, betac, sd=sqrt(epsilon1), log = T)
#log{q(candidate|current value)}
q2 <- dnorm(betac-derivbeta(beta) * epsilon1/2, beta, sd=sqrt(epsilon1), log = T)

D <- llhoodc + prhoodc - (llhood + prhood) + q1 - q2


u <- runif(1)

#posterior likelihood at candidate > posterior likelihood at current value
#=> exp(D) > 1 => D > 0

#When you generate u as runif(1), 'u' bounded [0,1]. log(u) < 0 for all possibilities of u.
#log(u) < D for any 'u', generated.

#D is small => exp(D) is small
#=> posterior likelihood at candidate < posterior likelihood at current value
#The condition log(u) < D will hold for smaller possibilities of u.

if(log(u) < D){ #u < exp(D) [=ratio of likelihoods] P(log(u)<D)=P(u<exp(D))=exp(D)
  beta <- betac
  llhood <- llhoodc
  prhood <- prhoodc

  flag1 <- flag1 + 1 #To count number of accepts
}

betals[[itr]] <- beta #storing the posterior samples

temp <- log(sigma) + derivsig(log(sigma))*epsilon2/2 + rnorm(1, sd = epsilon2)
sigmac <- exp(temp)

llhoodc <- sum(dnorm(y, X%*%beta, sigmac, log = T)) #log-likelihood of the data
prhoodc <- dnorm(beta, priormean, sd=sqrt(priorvar), log = T)+dgamma(1/sigmac^2, al0, be0) #prior l

#llhoodc + prhoodc is actually posterior log-likelihood for beta, sigmac
```

```
#log{q(current value|candidate)}
q1 <- dnorm(log(sigma)-derivsig(temp) * epsilon2/2, temp, sd=sqrt(epsilon2), log = T)
#log{q(candidate|current value)}
q2 <- dnorm(temp-derivsig(log(sigma)) * epsilon2/2, log(sigma), sd=sqrt(epsilon2), log = T)

D <- llhoodc + prhoodc - (llhood + prhood) + q1 - q2

u <- runif(1)

#posterior likelihood at candidate > posterior likelihood at current value
#=> exp(D) > 1 => D > 0

#When you generate u as runif(1), 'u' bounded [0,1]. log(u) < 0 for all possibilities of u.
#log(u) < D for any 'u', generated.

#D is small => exp(D) is small
#=> posterior likelihood at candidate < posterior likelihood at current value
#The condition log(u) < D will hold for smaller possibilities of u.

if(log(u) < D){ #u < exp(D) [=ratio of likelihoods] P(log(u)<D)=P(u<exp(D))=exp(D)
  sigma <- sigmac
  llhood <- llhoodc
  prhood <- prhoodc

  flag2 <- flag2 + 1 #To count number of accepts
}

sigmals[[itr]] <- sigma #storing the posterior samples

#For M-H sampler + univariate case, usually 0.45 is an optimal acceptance rate.

if(itr %% 100 == 0){ #Auto-tuning step
  ar <- flag1 / itr #Acceptance rate
  if(ar < 0.45){ #Acceptance is too low
    epsilon1 <- epsilon1 / 2
  }

  if(ar > 0.70){ #Acceptance is too high
    epsilon1 <- epsilon1 * 2
  }
  print(ar) #To track the acceptance rate

  ar <- flag2 / itr #Acceptance rate
  if(ar < 0.45){ #Acceptance is too low
```

```
    epsilon2 <- epsilon2 / 2
  }

  if(ar > 0.70){ #Acceptance is too high
    epsilon2 <- epsilon2 * 2
  }
  print(ar) #To track the acceptance rate
}
  }
  return(list(betap=betals,sigmap=sigmals))
}
```

Since it is not RW-MH, we required to compute the transition probability.

Running the chain and checking covergence

```
 samplesout <- chainseed(1, Total_itr = Total_itr)
betasamples  <- samplesout$betap[(burn+1):Total_itr]
sigmasamples <- samplesout$sigmap[(burn+1):Total_itr]

betas <- unlist(betasamples)
acf(betas) #Checking mixing by autocorrelation
acfvals <- acf(betas)

#Effective sample size. This number below essentially tells us our set of generated values
#correspond to how many independent samples.

M <- 2 #choice of M depends on acfvals$acf. abs(acfvals$acf) should always decrease.
#Then we choose M such that abs(acfvals$acf) is small enough (below th blue line in the plot)
#Here we choose 7 looking at the plot.

5000/(1+2*sum(acfvals$acf[1:M]))

#Depending on above ACF plot collect each 7-th sample

index <- (1:(5000/7))*7 - 6 #This is called thinning to reduce correlation

acf(betas[index]) #Re-checking mixing by autocorrelation

mean(betas[index]) #posterior mean
plot(density(betas[index])) #posterior density

quantile(betas[index], probs = c(0.025, 0.975)) #Equal tail 95% CI

library(coda)
```

```
chain <- mcmc(betas)

summary(chain)
plot(chain)

samplesout2 <- chainseed(2, Total_itr = Total_itr)

betasamples2  <- samplesout2$betap[(burn+1):Total_itr]
sigmasamples <- samplesout2$sigmap[(burn+1):Total_itr]

betasout2 <- unlist(betasamples2)

betas2 <- unlist(betasout2[(burn+1):Total_itr])
chain2 <- mcmc(betas2)

combinedchains = mcmc.list(chain, chain2)
plot(combinedchains)
gelman.diag(combinedchains)
gelman.plot(combinedchains)

library(BayesianTools)
correlationPlot(data.frame(chain))
```

2 Concluding remarks

Here is the list of things to consider:

- The support of the parameter: Depending on the support of the parameter, you need to decide whether you need a transformed parameter for the MH move. Note that it is better to have the parameter to be unrestricted. If the parameter is already unrestricted, you are good. Otherwise, you may revisit the above table to come up with an appropriate transformation.
- If a parameter is conditionally conjugate, it may be better to use Gibbs sampling than MH.

Part MCMC diagnostic

The theoretical convergence properties of MCMC (that the generated samples using MCMC will be able to approximate the true posterior) are based on some assumptions on the generated MCMC samples. These assumptions are stationarity, reversibility and ergodicity. Sufficient conditions for ergodicity are (1) irreducibility: the chain can reach each possible outcome whatever the starting position; (2) aperiodicity: there is no cyclic behavior in the chain and (c) positive

recurrence: the chain visits every possible outcome an infinite number of times and the expected time to return to a particular outcome, irrespective of where we start in the chain, is finite. In practical terms, ergodicity means that the chain will explore the posterior distribution exhaustively.

Convergence diagnostics thus essentially involve testing whether the generated samples satisfy those conditions: 1) stationarity, 2) reversibility and 3) ergodicity. Reversibility is automatically satisfied, as the expression for the acceptance probability rely on the detailed balance condition.

Detailed balance condition: Given a transition function, it is possible to define an acceptance probability $a(\theta \to \theta')$ that gives the probability of accepting a proposed mutation from θ to θ' in a way that ensures that the distribution of samples is proportional to $f(\theta)$, our target density, which in our context is $P(\theta \mid \mathbf{y})$. If the distribution is already in equilibrium, the transition density between any two states must be equal by detailed balance condition:

$$P(\theta \mid \mathbf{y})T(\theta \to \theta')a(\theta \to \theta') = P(\theta' \mid \mathbf{y})T(\theta' \to \theta)a(\theta' \to \theta)$$

. Note that in our previous definitions, $T(\theta \to \theta') = q(\theta' \mid \theta)$ and $T(\theta' \to \theta) = q(\theta \mid \theta')$.

The solution of $a(\theta \to \theta')$ that maximizes the rate at which above equilibrium is reached is $a(\theta \to \theta') = \min\{1, \frac{P(\theta'|\mathbf{y})q(\theta|\theta')}{P(\theta|\mathbf{y})q(\theta'|\theta)}\}$

Why Gibbs sampling is a special case of MH sampling with acceptance probability $a(\theta \to \theta') = 1$? To prove the above, let's go step by step.

- What is proposal distribution in case of a Gibbs sampler? (Meaning, how are we generating a candidate?)
- What is the transition probability? (q(Candidate | Current) and q(Current | Candidate)?)

There are plenty of diagnostics tools. We will just cover the most popular ones. MCMC chains share commonalities with time series datasets. Hence, most of the diagnostic tools are borrowed from exploratory analysis methods of time series datasets.

3 Trace plots

Trace plot is essentially to plot the postburn samples like `plot(lambdap)`. This will tell us to visually detect any pattern in the generated samples. Independent time series are also called "White noise". Our ultimate goal is to obtain independence posterior samples. Hence, the trace plot of the posterior samples should look like white noise.

4 Autocorrelation

MCMC samples by construction are autocorrelated. In case of Gibbs sampling, every alternate sample is expected to be independent. Hence, thinning by 2 is always kind of default. However,

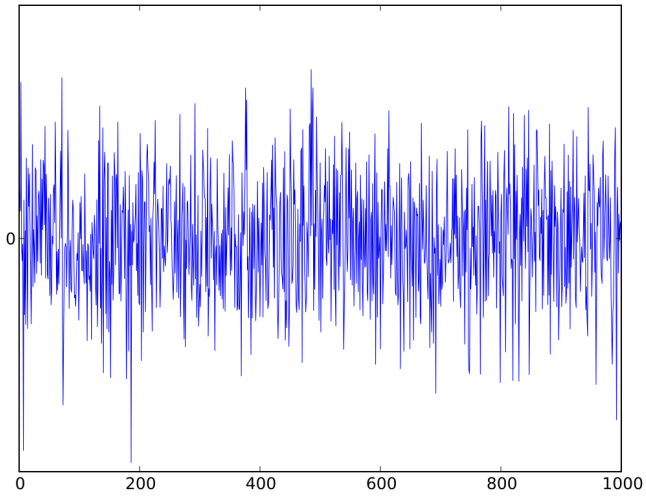

Figure 4: Typical trace plot of white noise and thus an ideal or desired plot of the MCMC samples.

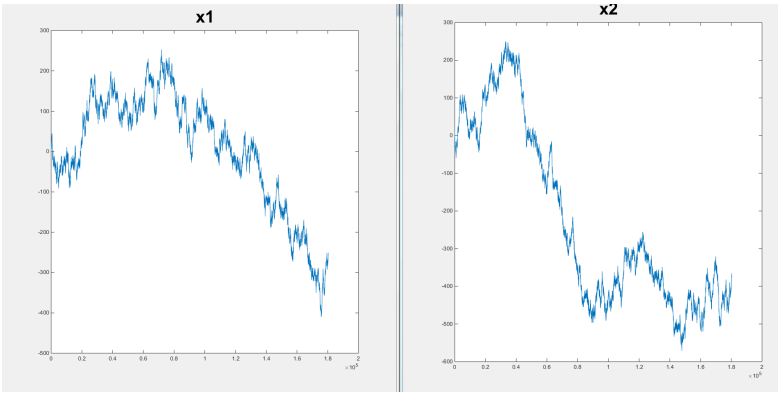

Figure 5: An example of bad convergence.

for general MH, it is not generally true. We need to make acf() plots to ascertain appropriate the lag beyond which there is no significant dependence.

The main motivation of the diagnostic test is to check whether there is any discernible pattern.

5 Diagnostic tests

Four diagnostic tests for assessing stationarity and/or accuracy are introduced here. The first three tests assess convergence on a single chain and are based on the time-series or stochastic process properties of a Markov chain. The fourth diagnostic evaluates the discrepancy between multiple Markov chains to detect nonstationarity. Two diagnostics evaluate stationarity (size of burn-in part κ_0) and accuracy (number of extra iterations κ_1).

Geweke diagnostic: This diagnostic test looks only for k0. Geweke (1992) suggests to formally test the stationarity of a Markov chain by comparing the means of an early and a late part of the chain using a (frequentist) significance test. If the n values θ were i.i.d. and split up into two different parts: A (early part) with n_A elements and B (late part) with n_B elements, then their respective (posterior) means $\bar{\theta}_A$ and $\bar{\theta}_B$ could be compared with a Z-test given by $Z = \frac{\bar{\theta}_A - \bar{\theta}_B}{s_A^2/n_A + s_B^2/n_B}$. Here s_A and s_B are sample standard deviations. However, the elements of the Markov chain are dependent. Hence, the two means and the standard deviations are all dependent. The means however are asymptotically unbiased. One can use spectral methods (using Fourier transformation) to obtain better estimates for the variances.

Geweke (1992) however suggested taking for A the initial 10% of the iterations ($n_A = n/10$) and for B the last 50% ($n_B = n/2$) to create a distance between the two parts. Then increase n so that the overall Z-test becomes insignificant at say, $\alpha = 0.05$. When the overall Z-test is significant at $\alpha = 0.05$, either the burn-in part (i.e. κ_0) was taken too small and/or the total chain is too short. A dynamic version of Geweke diagnostic might also help to find a better value for κ_0. For the dynamic version of the test, the Z-test is applied on $100(K-m)/K\%(m = 0, \ldots, K)$ last iterations of the chain. This produces $Z_m(m = 0, \ldots, K)$ test statistics that are plotted in a time-series plot.

Brooks–Gelman–Rubin (BGR) diagnostic: When the posterior is multi-modal, a single Markov chain might get stuck for a very long time in an area around a local mode. Convergence to a local mode is a well-known problem in classical optimization routines. To circumvent this problem it is advised to start up the optimization routine from various initial positions. In the same spirit, Gelman and Rubin (1992) suggested a convergence diagnostic based on multiple chains with 'overdispersed' (relative to the posterior distribution) starting positions. Their diagnostic is based on an ANOVA idea where the chains play the role of groups.

The algorithm first requires to run the MCMC with several starting values in parallel. The samples from a fixed MCMC chain can be regarded as a group. Hence, if we run M many parallel chains for a parameter θ and get the posterior samples $(\theta_{m,i})_{1 \le i \le n}$ for each $m = 1, \ldots, M$.

Geweke's diagnostic	`geweke.diag`
Heidelberger–Welch (HW) diagnostic	`heidel.diag`
Raftery–Lewis (RL) diagnostic	`raftery.diag`
Brooks–Gelman–Rubin (BGR) diagnostic	`gelman.diag`

Table 2: Diagnostic codes for different methods are available in R package `coda`. The required functions are listed above.

Here n stands for total number of samples. We can apply classical within- and between-group ANOVA analysis to check if there is any group difference. That's the main idea.

There are two more diagnostic tools. 1) Heidelberger–Welch (HW) diagnostic which kind of automates Geweke diagnostic. The above convergence diagnostics are all based on the posterior mean. There is 2) Raftery–Lewis (RL) diagnostic which implements a quantile based diagnostic measure, not posterior mean.

6 Effective sample size

This number essentially tells us our generated MCMC samples correspond to how many independent samples. It is given by (Total postburn samples)$/(1 + 2\sum_{j=1}^{M} |\rho_j|)$, where ρ_j is the correlation associated with j-th lag. In R outout `acf`, it is j+1-th values. Hence if $M = 7$ and (Total postburn samples) $= 5000$. It will be `(Total postburn samples)/(1+2*sum(acfvals$acf[2:(M+1)])).`

7 Nimble implementation of the car example and diagnostic tools

Nimble always implements MH sampling. It does not implement Gibbs. Hence, our manual implementation in many cases will be better than what Nimble does. In this example, we consider semi-conjugate prior, not conjugate NIG prior.

```
write('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', file = "~/.Renviron", append = TRUE)

install.packages("nimble", repos = "http://r-nimble.org", type = "source")

##############FOR WINDOWS the following part is extra
#########Install Rtools################ From https://cran.r-project.org/bin/windows/Rtools/

path <- Sys.getenv('PATH')
newPath <- paste("C:\\rtools40;C:\\rtools40\\bin;C:\\rtools40\\mingw_64\\bin;",
                 path, sep = "")
Sys.setenv(PATH = newPath)
```

```
library(nimble)

#Writing the model along with prior
code <- nimbleCode({
  #Define all the prior and other distributions
  for(k in 1:p)
    beta[k] ~ dnorm(0, sd = 100)
  sigma ~ dinvgamma(0.1,0.1)
  #sigma ~ dunif(0, 100)  # prior for variance components based on Gelman (2006)
  for(i in 1:n) {
    Y[i] ~ dnorm(inprod(beta[1:p], X[i, 1:p]), sd = sigma)
    #Y[i] ~ dnorm(beta[1]*X[i, 1]+beta[2]*X[i, 2] + inprod(beta[3:p], X[i, 3:p]), sd = sigma)
    #Both of the above are allowed. This is to show you different ways for better understanding
    #its usage.
  }
})

#The above block will be problem specific

## Car data######
data <- mtcars
Xmat <- cbind(data$wt,data$cyl, data$hp, data$am) # creating the desing matrix
y <- data$mpg

n <- nrow(Xmat)
d <- ncol(Xmat)

##############Estimation of model parameters start here

#Initialization: There are two parameters. Inital values are passed as a list()
fit <- lm(y~Xmat-1)
inits <- list(beta = array(fit$coefficients), sigma = 0.5)

constants <- list(n = n, p = d)
data <- list(
  Y = y,
  X = Xmat
)

## create the model object
#This part will be more or less the same for all the models
lmModel <- nimbleModel(code = code, constants = constants, data = data,
                       inits = inits, check = FALSE)
```

```
lmMCMC <- buildMCMC(lmModel)

ClmModel <- compileNimble(lmModel)

ClmMCMC <- compileNimble(lmMCMC, project = lmModel,showCompilerOutput = TRUE)

#Running MCMC samples
samples2 <- runMCMC(ClmMCMC, niter = 10000, nburnin = 0, nchains = 2)

samples <- samples2[[1]]
#Posterior samples of beta and sigma
betas   <- samples[, -5]
sigma2s <- (samples[, 5])^2 #squaring the samples to get samples of variance

#############Diagnostic tools################################
library(coda)
#Making the chain as MCMC object
chainsig <- mcmc(sigma2s)
chainbetF <- mcmc(betas)

geweke.diag(chainsig, frac1=0.1, frac2=0.5)
pnorm(abs(geweke.diag(mcmc(chainsig))$z),lower.tail=FALSE)*2

geweke.diag(chainbetF, frac1=0.1, frac2=0.5)
pnorm(abs(geweke.diag(mcmc(chainbetF))$z),lower.tail=FALSE)*2

#For BGR we need to run another chain
#Running MCMC samples
samples1 <- samples2[[2]]

#Posterior samples of beta and sigma
betas1   <- samples1[, -5]
sigma2s1 <- (samples1[, 5])^2 #squaring the samples to get samples of variance

chainsig1 <- mcmc(sigma2s1)
chainbet11 <- mcmc(betas1[, 1])

combinedchains = mcmc.list(chainsig, chainsig1)
plot(combinedchains)
gelman.diag(combinedchains)
gelman.plot(combinedchains)

combinedchains = mcmc.list(chainbet1, chainbet11)
```

```
plot(combinedchains)
gelman.diag(combinedchains)
gelman.plot(combinedchains)

chainbetF <- mcmc(betas)
chainbetF1 <- mcmc(betas1)
combinedchains = mcmc.list(chainbetF, chainbetF1)
plot(combinedchains)
gelman.diag(combinedchains)
gelman.plot(combinedchains)

#The 'potential scale reduction factor' is calculated for each
#variable in x, together with upper and lower confidence limits.
#Approximate convergence is diagnosed when the upper limit is close to 1.
#For multivariate chains, a multivariate value is calculated that bounds
#above the potential scale reduction factor for any linear combination of
#the (possibly transformed) variables.

#Not so important
heidel.diag(chainsig, eps=0.1, pvalue=0.05)

heidel.diag(chainbetF, eps=0.1, pvalue=0.05)

raftery.diag(chainsig, q=0.025, r=0.005, s=0.95, converge.eps=0.001)

raftery.diag(chainbetF, q=0.025, r=0.005, s=0.95, converge.eps=0.001)

###########Posterior summaries#######################
quantile(sigma2s, probs=c(0.025, 0.975)) #Credible intervals
apply(betas, 2, quantile, probs=c(0.025, 0.975))

#posterior means
mean(sigma2s)
apply(betas, 2, mean)

#Trace plot
plot(sigma2s, type='l')
plot(betas[, 1], type='l') #First coefficient of beta
plot(betas[, 3], type='l') #Third coefficient of beta

#acf plots
acf(sigma2s)
acf(betas[, 1])
```

```
acfvalssig  <- acf(sigma2s[5001:10000])
#acfvalsbet1 <- acf(betas[5001:10000, 1])

M <- 12

5000/(1+2*sum(acfvalssig$acf[2:(M+1)]))
```

Hierarchical Bayes

Hierarchical modeling is one of the those areas in statistics where Bayesians thrive. These are those models where we specify the statistical model in several layers. Different layers are motivated to capture different sources of noise. The most common example of Hierarchical modeling is the mixed effect model.

Say we are collecting longitudinal data on blood pressure (BP) of patients, and we are interested to study association between BP and BMI. Every patient will be visiting the clinic after every 3 months for a year. Hence, there are a total of 5 visits for each patient, visit0, visit3, visit6, visit9 and visit12. Let $y_{i,j}$ be the BP rating for i-th patient at j-th visit where $j = 0, 3, 6, 9, 12$ and $x_{i,j}$ stands for the BMI rating for i-th patient at j-th visit.

Using a simple regression model, we can write $\mathbb{E}(y_{i,j}) = \beta x_{i,j}$. Hence, an appropriate model may be $y_{i,j} = \beta x_{i,j} + \epsilon_{i,j}$ for $\epsilon_{i,j} \sim \text{Normal}(0, \sigma^2)$. [Note that I write $y_{i,j} = \beta x_{i,j} + \epsilon_{i,j}, \quad \epsilon_{i,j} \sim \text{Normal}(0, \sigma^2)$ and $y_{i,j} \sim \text{Normal}(\beta x_{i,j}, \sigma^2)$ alternatively as they mean the same thing.] Is this an adequate model?

Well under this model $\text{cov}(y_{i,0}, y_{i,3}) = 0$ or more generally $\text{cov}(y_{i,j}, y_{i,j'}) = 0$ for any $j \neq j'$. This is unreasonable, as $y_{i,0}$ and $y_{i,3}$ both correspond to the same subject. Thus $y_{i,0}$ and $y_{i,3}$ are expected to be correlated. To introduce this correlation, we bring in another source of noise via random effect and finally write the complete model as,

$$\begin{aligned} &\text{Level 1: } y_{i,j} \sim \text{Normal}(\beta x_{i,j} + b_i, \sigma^2). \\ &\text{Level 2: } b_i \sim \text{Normal}(0, \sigma_1^2) \end{aligned}$$

Now $\text{cov}(y_{i,0}, y_{i,3}) = \sigma_1^2$. One may argue that the covariance is not general enough, as it introduces the same level of correlation for all the pairs of observations. However, the generalization is easy. For simplicity, we will not discuss the generalization.

Finally, the priors:

$$\text{Priors: } \beta \sim \text{Normal}(0, 100); \quad \sigma^{-2}, \sigma_1^{-2} \sim \text{Ga}(0.1, 0.1)$$

Steps towards Bayesian inference:

- Likelihood: Note that the distributions at Layer 1 and Layer 2 are independent.

- Sampling: In addition to the mainstream parameters β, σ, σ_1, we have the random effects b_i in our model. These are also unknown. In Bayesian framework, we treat anything unknown as parameters and thus consider sampling.

References

[1] Jan Hannig, Hari Iyer, Randy CS Lai, and Thomas CM Lee. Generalized fiducial inference: A review and new results. *Journal of the American Statistical Association*, 111(515):1346–1361, 2016.

[2] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[3] Emmanuel Lesaffre and Andrew B Lawson. *Bayesian biostatistics*. John Wiley & Sons, 2012.