# QUEEN MARY, UNIVERSITY OF LONDON

## MTH6102: Bayesian Statistical Methods

### Practical 8

### 2023-2024

## 1 Uninformative prior distributions

In lecture week 6, when considering uninformative prior distributions, it was stated that uninformative flat prior distributions are not invariant under nonlinear monotone transformations. For example, if the probability distribution for $\sigma$ is uniform or flat on the interval $[0, c]$ for some $c > 0$, then the pdf for $\sigma^2$ is not uniform. It is, in fact, proportional to

$$\frac{1}{2\sqrt{\sigma^2}}.$$

The pdf for $\sigma$ is

$$f(\sigma) = \frac{1}{c},\ 0 \leq \sigma \leq c.$$

In full, if we put $\gamma = \sigma^2$, then the pdf for $\gamma$ is

$$g(\gamma) = \frac{1}{2c\sqrt{\gamma}},\ 0 \leq \gamma \leq c^2.$$

To check this, for some $c > 0$, generate a random sample for $\sigma$ of size $10,000$ using the command `runif` and save it in the object `sigma`. Then generate a sample for $\gamma$ by transforming the $\sigma$ sample:

```
sigma = runif(10000)
gamma = sigma^2
```

Draw a histogram of the sample `gamma`, using the command

```
hist(gamma, freq=FALSE)
```

The option `freq=FALSE` ensures that the histogram is on the probability density scale.

Plot the pdf $g(\gamma)$, for example using the `curve` or `plot` commands. The histogram and the curve should be roughly the same shape.

Similarly, if $\theta = \log(\sigma)$, then the pdf for $\theta$ is

$$h(\theta) = \frac{e^\theta}{c},\ -\infty \leq \theta \leq \log(c).$$

Generate a random sample for $\theta$, and check the pdf against a histogram.

## 2 Checking prior distributions

In the 8A lecture, we looked at some examples of choosing a (conjugate) prior distribution based on some previous information. It is possible to check the derivations using simulation.

For example, suppose we want to choose the parameters of a Gamma$(\alpha, \beta)$ prior distribution to match a certain mean and standard deviation, $m$ and $s$, respectively. Take $m = 1$ and $s = 0.03$, and calculate the values of $\alpha$ and $\beta$ that are needed to match $m$ and $s$, as explained in lecture 8A.

To check the calculations, we could generate a sample from this gamma distribution

```
v = rgamma(10000, shape=alpha, rate=beta)
```

Having done this, then find the mean and standard deviation of `v` to check that they are approximately equal to $m$ and $s$.

Now assuming a prior gamma distribution for $\lambda$, take $\alpha = 1$ and we want to find $\beta$ such that (for some known $c > 0$)

$$P(\lambda > c) = 0.1$$

For $c = 2$, find the value of $\beta$ that is needed. Note that when $X \sim \text{gamma}(1, \beta)$, then $X \sim \text{exponential}(\beta)$. The cdf of $X \sim \text{exponential}(\beta)$ is $F_X(c) = P(X \leq c) = 1 - \exp(-\beta c)$, $c \in \mathbb{R}$.

Then use the `pgamma` command to check the calculation, using code such as

```
pgamma(c, shape=alpha, rate=beta)
```

Also generate a random sample `w` from this gamma distribution, and count what proportion of the sample is greater than `c` using the command

```
mean(w>c)
```

## 3 Simple Monte Carlo for comparing two binomial probabilities

We looked at this problem in lecture 5B as an example of simulating from a joint posterior for the Binomial case.

Suppose that the data are the outcome of a clinical trial of two treatments for a serious illness, the number of deaths after each treatment. Let the data be $k_i$ deaths out of $n_i$ patients, $i = 1, 2$ for the two treatments, and the two unknown parameters are $q_1$ and $q_2$, the probability of death with each treatment.

Assuming a flat prior $p(q_1, q_2) = 1$, we found that posterior distributions are independent, and

$$p(q_1 \mid k_1) \sim \text{beta}(k_1 + 1, n_1 - k_1 + 1), \quad p(q_2 \mid k_2) \sim \text{beta}(k_2 + 1, n_2 - k_2 + 1)$$

Let

$$\psi = \log\left(\left(\frac{q_1}{1 - q_1} \div \frac{q_2}{1 - q_2}\right)\right)$$

be the log-odds ratio. We would like to estimate $\psi$ using Monte Carlo integration.

Take the observed data to be $n_1 = 240, k_1 = 15$; $n_2 = 280, k_2 = 12$, and simulate two vectors in R, each of length $10,000$, containing random samples from the posterior distributions of $q_1$ and $q_2$

```
Nsim = 10000
q_sim1 = ...
q_sim2 = ...
```

Generate a posterior sample for the difference $\psi$, using the two vectors already generated.

```
odds1=q_sim1/(1-q_sim1)
odds2=q_sim2/(1-q_sim2)
psi=log(odds1/odds2)
```

Use this sample to calculate the posterior mean for $\psi$. The posterior mean for $\psi$ is the Monte Carlo integration estimator of $\psi$.

Compute an equal tail 95% credible interval for $\psi$ using this command

```
quantile(psi, probs=c(0.5, 0.025, 0.975))
```

Note that `psi` contains independent observations of $\psi$ and is a sample from $p(\psi \mid k_1, k_2)$ the posterior density of $\psi$. Plot the histogram of the posterior density of $\psi$.

# 4 Appendix

### Simple graph options

We have used named colours such as red and blue in some examples, but there are many more available. The R command `colors()` lists them, while there are documents elsewhere on the web that show them visually, which is more helpful. One is here.

For example, to draw a histogram and choose the colour of the bars:

```
hist(iris$Petal.Length, col="papayawhip")
```

Some other useful options are the following, which are illustrated in examples below. They can found among the many options in `?par`.

- `lwd` - line width, default is 1;
- `lty` - line type, e.g. `"dashed"`, `"dotted"`;
- `pch` - plotting character;
- `main` - graph title;
- `xlab, ylab` - axis titles;
- `xlim, ylim` - limits of axis ranges;
- `legend` - add a legend.

```
x = 1:8
y = 3 + 2*x + 0.4*rnorm(length(x))
z = y + 1
plot(x=x, y=y, xaxs="r", yaxs="r", pch=15, col="royalblue")
points(x=x, y=z, pch=17, col="seagreen")
```

The following two lines show the symbols available for plotting points:

```
xp=0:25
plot(x=xp, y=rep(1, length(xp)), pch=xp)
```

A legend can be added, but it does not automatically pick up elements such as colour and line type from the graph, so these need to be specified both in the plot and the legend:

```
plot(dnorm, xlim=c(-3, 3), ylim=c(0, 0.5), lwd=2, col="slateblue",
    main="Normal pdf", xlab="y", ylab="pdf")
x = seq(from=-3, to=3, by=0.1)
y = dnorm(x, mean=0, sd=0.8)
lines(x=x, y=y,  lty="dashed", lwd=2, col="red3")
legend("topright", legend=c("sd = 1", "sd = 0.8"),
    col=c("slateblue", "red3"), lty=c("solid", "dashed"), lwd=c(2, 2))
```

**Saving graphs**

If you are using Microsoft Word, then just copying graphs from R into your document may be all you need. But for other purposes, it can be useful to save graphs to disk. For use in a Latex document, saving the graph as a pdf is a good option. The second line can be replaced with multiple lines for drawing the graph.

```
pdf("norm.pdf", width=5, height=4)
plot(dnorm, xlim=c(-3, 3), col="seagreen3")
dev.off()
```

This will save the file to our current working directory, which we can check using `getwd()` and change using `setwd(...)`.

`?Devices` lists ways of saving in different graph formats.