The Problem: A new treatment is given to 100 patients. Of them, only 8 respond. But there is a subgroup of 5 in which 3 patients respond, yielding a response rate of 60%! Should the treatment be recommended (or at least developed further) for people in the subgroup?

	Dark Hair (D)	**Light Hair (L)**	TOTAL
Responder (R)	**3**	**5**	8
Nonresponder (N)	**2**	**90**	92
TOTAL	5	95	100

We want to express prior knowledge or belief concerning how related $P_{R.D}$ and $P_{R.L}$ are.
Dr. Lump is sure that they are identical; Dr. Lump lumps together the *D* and *L* groups.
Dr. Split is sure that they are totally unrelated.
Others may have intermediate positions.
Or, how close to splitting or lumping seem appropriate will depend on the nature of *X*.

Let's transform the two conditional probabilities with the log odds function.

For *X*= *D* or *L*,

$$z_{R.X} = \mathrm{logit}(\Pr(R\,|\,X)) = \log\left(\frac{\Pr(R\,|\,X)}{1-\Pr(R\,|\,X)}\right)$$

Then a broad family of useful priors is bivariate normal in the (logit($P_{R.D}$), logit($P_{R.L}$)) plane:

$$(z_{R.D}, z_{R.L})^T \sim N(\mu_z, \Sigma_z) \text{ where } \mu_z = (\mu_{R.D}, \mu_{R.L})^T \text{ and } \Sigma_z = \begin{pmatrix} \tau+\phi & \tau \\ \tau & \tau+\phi \end{pmatrix}.$$

The prior means for the logits are $\mu_{R.D}, \mu_{R.L}$. The superscript "*T*" means "transpose"; customarily these vectors are written as column vectors. (We usually set the prior means of $z_{R.D}$ and $z_{R.L}$ to be the same, so $\mu_{R.D} = \mu_{R.L}$.) Any factors that might affect both response probabilities are represented by the covariance τ. Any that are not shared are represented by $\phi = (\tau+\phi) - \tau$. The prior correlation between $z_{R.D}$ and $z_{R.L}$ is $\tau/(\tau+\phi)$.

So the joint density of (logit($P_{R.D}$), logit($P_{R.L}$)) is

$$f_z(z_{R.D}, z_{R.L}) = (2\pi)^{-1/2} \det(\Sigma_z)^{-1/2} \exp\left(-\frac{1}{2}(z-\mu_z)^T \Sigma_z^{-1}(z-\mu_z)\right)$$

which can be converted to a density on the original scale by multiplying by the Jacobean of the logit transformation:

$$f_P(P_{R.D},P_{R.L}) = f_z(z_{R.D},z_{R.L}) \times \det\begin{pmatrix} \partial z_{R.D}/\partial P_{R.D} & 0 \\ 0 & \partial z_{R.L}/\partial P_{R.L} \end{pmatrix}^{-1}$$
$$= f_z(\operatorname{logit} P_{R.D}, \operatorname{logit} P_{R.L}) \times P_{R.D}(1-P_{R.D}) \times P_{R.L}(1-P_{R.L})$$

because the Jacobean of the transformation is

$$\frac{d}{dP}z = \frac{d}{dP}\left(\log((P/(1-P))\right) = \frac{1}{P/(1-P)}\frac{d}{dP}(1-1/(1-P))$$
$$= \frac{1-P}{P}(1-P)^{-2} = P^{-1}(1-P)^{-1} = P^{-1} + (1-P)^{-1}$$

We use $f_P(P_{R.D},P_{R.L})$ to draw the contours of prior and posterior distributions for $\boldsymbol{P}$.

The model (conditional distribution of the data):

The observed proportions are also converted to logits: $x_{R.X} = \operatorname{logit} \hat{P}_{R.X}$, and the variances of these logits is estimated with the delta method applied to the logit function.

$$\operatorname{var}(\operatorname{logit}(\hat{\boldsymbol{P}})) \doteq \operatorname{var}(\hat{\boldsymbol{P}})\left(\frac{\boldsymbol{d}}{\boldsymbol{dP}}\operatorname{logit}\boldsymbol{P}\right)^2$$
$$= \boldsymbol{n}^{-1}\boldsymbol{P}(1-\boldsymbol{P})\left(\boldsymbol{P}^{-1}(1-\boldsymbol{P})^{-1}\right)^2 = \boldsymbol{n}^{-1}\left(\boldsymbol{P}^{-1}(1-\boldsymbol{P})^{-1}\right)$$
$$\doteq \boldsymbol{n}^{-1}\left(\hat{\boldsymbol{P}}^{-1}(1-\hat{\boldsymbol{P}})^{-1}\right)$$

When there is a zero count, the variance formula yields infinity.
A small "fudge factor" can be added to each count, to produce a finite variance.

We model the data, conditional on the unknown $z_{R.D}, z_{R.L}$ (which is what we are interested in knowing), with a normal approximation for the logits:

$$\begin{pmatrix} x_{R.D} \\ x_{R.L} \end{pmatrix} \Bigg| \begin{pmatrix} z_{R.D} \\ z_{R.L} \end{pmatrix} \sim N\left(\begin{pmatrix} z_{R.D} \\ z_{R.L} \end{pmatrix}, \Sigma_{x|z}\right)$$

$$\text{where } \Sigma_{x|z} = \begin{pmatrix} n^{-1}P_{R.D}^{-1}(1-P_{R.D})^{-1} & 0 \\ 0 & n^{-1}P_{R.L}^{-1}(1-P_{R.L})^{-1} \end{pmatrix}.$$

(Notice the very tall "|" indicating conditioning on the z 's.) Conditionally, the data are independent, as we see from the off-diagonal zeros.

The joint distribution

The joint distribution of the unknown z's and the observed x's is the product of the prior distribution and the model. This is a good spot to simplify our notation.

$$Y_1 = \begin{pmatrix} z_{R.D} \\ z_{R.L} \end{pmatrix}, \quad Y_2 = \begin{pmatrix} x_{R.D} \\ x_{R.L} \end{pmatrix}, \quad \mu_1 = E(Y_1) = \mu_2 = E(Y_2) = \begin{pmatrix} \mu_{R.D} \\ \mu_{R.L} \end{pmatrix}$$

$$\Sigma_{11} = \mathrm{var}(Y_1) = \Sigma_z = \begin{pmatrix} \tau+\phi & \tau \\ \tau & \tau+\phi \end{pmatrix}, \quad \Sigma_{22} = \mathrm{var}(Y_2) = \Sigma_x = \Sigma_z + \Sigma_{x|z}$$

$$\Sigma_{12} = \Sigma_{21} = \mathrm{cov}(Y_1, Y_2) = \Sigma_z$$

Then the joint distribution of the Y's (in our case, z and x) is

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right)$$

and the beautiful thing is that the posterior distribution we want is the conditional distribution of the unknown Y_1 (which is z) conditional on the observed Y_2 which is (x). We get this from the standard formulas for conditional mean and variance for the multivariate normal distribution:

$$Y_2 \middle| Y_1 = y_1 \sim N\left(\mu_{2.1}, \Sigma_{22.1}\right)$$

where

$$\mu_{2.1} = E(Y_2 \mid Y_1 = y_1) = \mu_2 + \Sigma_{21} \Sigma_{11}^{-1}(y_1 - \mu_1)$$

and

$$\Sigma_{22.1} = Var(Y_2 \mid Y_1 = y_1) = \Sigma_{22} - \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} .$$

Here it is in code. It's not hard to do. The function `solve`() does the matrix inversion.

```
varhat = apply(DLdata, 1, function(r)sum(1/r))
sig11 = sig12 = sig21 = matrix(c(tau+phi,tau,tau,tau+phi),nrow=2)
sig22 = sig11 + diag(varhat)   ## marginal variance of the data
logit.prior.mean = c(mu0, mu0)
## always use fudged here:
postmean.logit = logit.prior.mean +
  sig12%*%solve(sig22) %*% (logit.hat.fudged-logit.prior.mean)
postmean.p = antilogit(postmean.logit)
postvar.logit = sig11 - sig12%*%solve(sig22)%*%sig21
```

To draw the contour plots, here is the code:

https://github.com/professorbeautiful/T15lumpsplit/blob/master/inst/T15lumpsplit/Plight-Pdark-posterior-new.R