# Answers to Odd-Numbered Exercises for Chapter 25 Fox, *Applied Regression Analysis and Generalized Linear Models, Third Edition* (Sage, 2016)

John Fox

Answers last modified: 2021-11-29

This document provides worked-out answers to the odd-numbered exercises in bonus Chapter 25 on Bayesian estimation of regression models (excluding the data-analysis exercises on the website for the text). The answers to both odd- and even-numbered questions, are available from the author for instructors using the text in a college or university class.

Although the text is written to be software-neutral, some of the exercises require the use of statistical software. I employed the R statistical computing environment (R Core Team, 2021) for these exercises, but you should feel free to substitute other appropriate statistical software for R. See Fox and Weisberg (2019) for an introduction to R in the context of regression analysis that largely overlaps with the text.

# Exercises for Chapter 25

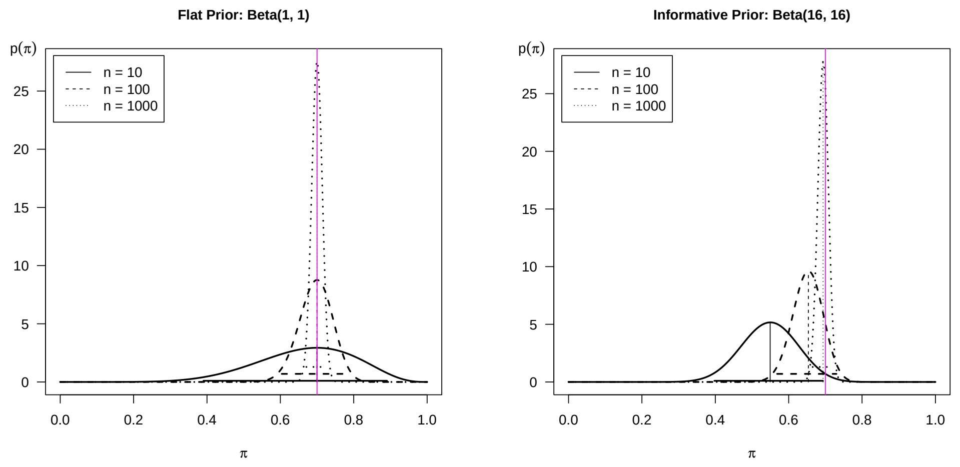
## Exercise 25.1

According to Bayes's theorem, $\Pr(A|P) = \Pr(P|A)\Pr(A)/\Pr(P)$, where $\Pr(P) = \Pr(P|A)\Pr(A) + \Pr(P|\overline{A})\Pr(\overline{A}) = 1 \times .1 + .1 \times .9 = .19$, and so $\Pr(A|P) = (1 \times .1)/.19 = .52$, which, indeed, is about half of $\Pr(P|A) = 1$.

To understand this result (and to take the hint in the problem), imagine a population in which no one has antibodies, so that instead of $\Pr(A) = .1$ (i.e., small), $\Pr(A) = 0$. Then anyone who tests positive is necessarily a false positive, because no one is really positive, and $\Pr(A|P) = 0$, as Bayes's theorem tells us for this case. Thus $\Pr(A|P)$ depends on the prevalence $P(A)$ of antibodies in the population as well as on the sensitivity and specificity of the test.

## Exercise 25.3

**(a)** Here are graphs of the posterior distribution of $\pi$ for the flat $\text{Beta}(a = 1, b = 1)$ prior (on the left) and the informative $\text{Beta}(a = 16, b = 16)$ prior (on the right):



In both cases, the solid lines show the posterior distribution, the posterior mode (vertical line), and 95% central posterior interval (horizontal line) for $n = 10$, the broken lines for $n = 100$, and the dotted lines for $n = 1000$. The vertical magenta line is at the proportion of heads in the data, which is .7 in all cases. Each posterior distribution is $\text{Beta}(h + a, n - h + b)$, where $h$ is the number of heads in the sample (e.g., 7 when $n = 10$).

For the flat prior, the posterior mode is always equal to .7 regardless of the sample size, but as the sample size grows, the posterior distribution becomes less spread out and thus the 95% posterior interval gets narrower. The informative prior pulls the posterior distribution towards its center at $\pi = .5$, but as the sample size grows, the influence of the prior weakens and by $n = 1000$, the posterior distribution is centered very close to $\pi = .7$; as well, as $n$ increases, the spread of the posterior distribution decreases and the 95% posterior interval gets narrower.

**(b)** As in part (a), the posterior distribution of $\pi$ is $\text{Beta}(h + a, n - h + b)$, but now $n = 10, a = 16$ and $b = 16$, while the number of heads $h$ varies from 0 to 10; the corresponding posterior modes are

| $h$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Posterior mode | .375 | .400 | .425 | .450 | .475 | .500 |
| $h$ | 6 | 7 | 8 | 9 | 10 | |
| Posterior mode | .525 | .550 | .575 | .600 | .625 | |

When the sample proportion ($h/10$) is below .5, so is the posterior mode, but because the sample size $n = 10$ is small, the posterior mode is drawn strongly upwards towards the center of the informative prior distribution at $\pi = .5$. Similarly, when the sample proportion is above .5, the posterior mode is drawn downwards towards $\pi = .5$.

**Exercise 25.5**

**(a)** $\sigma^2/n$ is the sampling variance of the mean of an independent random sample of size $n$.

**(b)** Precision is the inverse of variance. Consequently, the precision of the sample mean is the inverse of its sampling variance, and the precision of the prior mean is the inverse of the prior variance. Variance is a measure of dispersion or variation; if, for example, the sampling variance of the sample mean is large, then an estimate of $\mu$ based on the sample mean will be imprecise—will vary a great deal from sample to sample. The inverse of the variance of course increases as the variance declines, and thus is a measure of precision in the ordinary sense of the word.

In Bayesian estimation, there are two sources of information about $\mu$, one coming from the data (the sample mean $\bar{y}$), and the other from the prior (the prior mean $\mu_0$). As the precision of each of these sources of information increases (i.e., as the corresponding variance declines), the amount of information about $\mu$ increases—as it turns out, proportionally—and so it makes sense to weight each value by its precision to form a combined estimate of $\mu$.

**(c)** Combining information about $\mu$ from the data and the prior should produce a better estimate than using either source of information alone, and so it's sensible that the precision of the posterior mean, which combines the data with the prior, should be the sum of the two precisions and hence larger than either. Because the variance is the inverse of the precision, the posterior variance is smaller than the variance of the sample mean or the prior variance.

**Exercise 25.7**

I was able to obtain a usable solution by tightening the priors for the covariance components in the logit part of the model (although the resulting effective sample sizes for these covariance components are still relatively small), decreasing the standard deviations of the normal priors by a factor of 2: $\log_e \psi_{Z_j} \sim N(0, 0.2^2)$ for $j = 0, 1$, and $z(\rho_{Z_0 Z_1}) \sim N(0, 0.3^2)$. On the one hand, as indicated in the problem and in the text, adjusting the priors based on the results violates the Bayesian paradigm. On the other hand, the originally specified priors had no deep justification and the modified priors are still quite vague.

```
> library("rstan")
. . .
> (ncores <- parallel::detectCores())
[1] 20
> options(mc.cores=ncores) # for parallel computations
> rstan_options(auto_write=TRUE)

> data(Blackmore, package="carData")
> Blackmore$nonzero  <- as.numeric(Blackmore$exercise > 0)
> data.blackmore <- with(Blackmore,
+        list(n = nrow(Blackmore),
+             J = nlevels(subject),
+             subject = as.numeric(subject),
+             patient = as.numeric(group == "patient"),
+             age = age - 8,
+             log_exercise = ifelse(exercise == 0, 0, log2(exercise)),
+             nonzero = as.numeric(exercise > 0),
+             n_nonzero = sum(nonzero),
+             index_nonzero = which(as.logical(nonzero)))
+ )

> blackmore.model <- "
+     data {
```

```
+           int<lower=0> n; // total number of observations
+           int<lower=0> J; // number of subjects
+           int subject[n]; // subject index
+           vector[n] patient; // patient dummy regressor
+           vector[n] age; // within-subject predictor
+           vector[n] log_exercise; // response variable
+
+           int<lower=0, upper=1> nonzero[n]; // exercise > 0?
+           int n_nonzero;  // number of nonzero observations
+           int<lower=1, upper=n>
+               index_nonzero[n_nonzero]; // indices of nonzero observations
+       }
+       parameters {
+           matrix[J, 2] B; // subject-specific intercepts and slopes (deviations)
+           vector[4] beta; // group-level coefficients
+           real log_sigma; // error standard deviation
+           vector[2] log_psi_b; // standard deviations of coefficients
+           real z_rho_b; // Fisher z of correlation of subject-specific coefficients
+           matrix[J, 2] Z; // subject-specific intercepts and slopes
+                                  //      for logit part of model (deviations)
+           vector[4] zeta; // group-level coefficients for logit part of model
+           vector[2] log_psi_z; // standard deviations of subject-specific
+                                  //      logit coefficients
+           real z_rho_z; // Fisher z of correlation of subject-specific
+                                //       logit coefficients
+       }
+       transformed parameters {
+           // conditional expection for linear part of model
+           vector[n] mu = beta[1] + B[subject, 1] + beta[2]*patient + beta[3]*age
+                    + beta[4] * patient .* age + B[subject, 2] .* age;
+           // linear predictor for for logit part of model
+           vector[n] eta = zeta[1] + Z[subject, 1] + zeta[2]*patient + zeta[3]*age
+                            + zeta[4] * patient .* age + Z[subject, 2] .* age;
+           real sigma = exp(log_sigma);
+           vector[2] psi_b = exp(log_psi_b); // standard deviations of
+                                          //       subject-specific  coefficients
+           vector[2] psi_z = exp(log_psi_z); // standard deviations of
+                                          //      subject-specific logit coefficients
+           matrix[2, 2] Psi_b; // covariance matrix of subject-specific coefficients
+           matrix[2, 2] Psi_z; // covariance matrix of subject-specific  coefficients
+           real rho_b = tanh(z_rho_b); // correlation between subject-specific
+                                 //      intercepts and slopes
+           real rho_z = tanh(z_rho_z); // correlation between subject-specific logit
+                                 //      intercepts and slopes
+           Psi_b[1, 1] = psi_b[1]^2;
+           Psi_b[2, 2] = psi_b[2]^2;
+           Psi_b[1, 2] = psi_b[1]*psi_b[2]*rho_b;
+           Psi_b[2, 1] = psi_b[1]*psi_b[2]*rho_b;
+           Psi_z[1, 1] = psi_z[1]^2;
+           Psi_z[2, 2] = psi_z[2]^2;
+           Psi_z[1, 2] = psi_z[1]*psi_z[2]*rho_z;
+           Psi_z[2, 1] = psi_z[1]*psi_z[2]*rho_z;
+       }
+       model{
+           nonzero ~ bernoulli_logit(eta); // logit part of model
+           log_exercise[index_nonzero]
+              ~ normal(mu[index_nonzero], sigma); // linear part of model
+
+           for (j in 1:J){
```

```
+             B[j, ] ~ multi_normal(rep_vector(0, 2), Psi_b);
+             Z[j, ] ~ multi_normal(rep_vector(0, 2), Psi_z);
+                              // distribution of subject-specific coefficents
+         }
+
+         // priors:
+             // for linear part of the model:
+         beta[1] ~ normal(0, 0.5); // control intercept (index starts at 1, not 0)
+         beta[2] ~ normal(0, 0.5); // difference in intercept
+         beta[3] ~ normal(0, 0.5); // control age slope
+         beta[4] ~ normal(0, 1); // difference in slope
+         log_sigma ~ normal(0, 0.4);
+         log_psi_b ~ normal(0, 0.4);
+         z_rho_b ~ normal(0, 0.6);
+             // for logit part of the model:
+         zeta[1] ~ normal(0, 1); // control intercept
+         zeta[2] ~ normal(0, 0.5); // difference in intercept
+         zeta[3] ~ normal(0, 0.5); // control age slope
+         zeta[4] ~ normal(0, 0.5); // difference in slope
+         log_psi_z ~ normal(0, 0.2);
+         z_rho_z ~ normal(0, 0.3);
+     }
+ "

> system.time(fit.blackmore <- stan(model_code=blackmore.model, chains=4,
+     model_name= "Hurdle model for Blackmore data,
              not so weak priors, random intercepts and slopes",
+     seed=234870879, data=data.blackmore, iter=10000))
. . .
   user  system elapsed
 86.655   4.568 514.230

> print(fit.blackmore,
+       pars=c("beta", "zeta", "sigma", "psi_b", "rho_b", "psi_z", "rho_z"),
+       digits=4)
Inference for Stan model: Hurdle model for Blackmore data,
        not so weak priors, random intercepts and slopes.
4 chains, each with iter=10000; warmup=5000; thin=1;
post-warmup draws per chain=5000, total post-warmup draws=20000.

            mean se_mean     sd    2.5%     25%     50%     75%   97.5%
beta[1]  -0.2097  0.0020 0.1545 -0.5146 -0.3141 -0.2087 -0.1057  0.0961
beta[2]  -0.1998  0.0025 0.1956 -0.5841 -0.3304 -0.1999 -0.0689  0.1886
beta[3]   0.1040  0.0003 0.0276  0.0499  0.0853  0.1040  0.1228  0.1579
beta[4]   0.2113  0.0003 0.0343  0.1437  0.1883  0.2117  0.2344  0.2787
zeta[1]   2.3355  0.0040 0.2860  1.8039  2.1380  2.3287  2.5270  2.9192
zeta[2]  -0.3677  0.0023 0.3038 -0.9619 -0.5706 -0.3656 -0.1669  0.2248
zeta[3]   0.3156  0.0023 0.1196  0.0939  0.2337  0.3101  0.3929  0.5607
zeta[4]   0.2087  0.0014 0.1280 -0.0371  0.1214  0.2072  0.2954  0.4615
sigma     1.0206  0.0004 0.0347  0.9556  0.9963  1.0200  1.0441  1.0903
psi_b[1]  1.3553  0.0012 0.0936  1.1781  1.2908  1.3531  1.4169  1.5458
psi_b[2]  0.1519  0.0004 0.0196  0.1138  0.1388  0.1517  0.1649  0.1912
rho_b    -0.4825  0.0014 0.0998 -0.6591 -0.5527 -0.4889 -0.4200 -0.2682
psi_z[1]  1.1691  0.0065 0.2068  0.8122  1.0210  1.1535  1.2982  1.6148
psi_z[2]  0.5227  0.0019 0.0738  0.3934  0.4704  0.5180  0.5687  0.6800
rho_z     0.2019  0.0057 0.2018 -0.2001  0.0647  0.2061  0.3429  0.5834
          n_eff   Rhat
beta[1]    5920 1.0009
beta[2]    5928 1.0005
```

```
beta[3]   10241 1.0000
beta[4]   10247 1.0000
zeta[1]    5205 1.0006
zeta[2]   17623 1.0000
zeta[3]    2787 1.0032
zeta[4]    8860 1.0002
sigma      8469 1.0007
psi_b[1]   6194 1.0004
psi_b[2]   2329 1.0033
rho_b      4951 1.0006
psi_z[1]   1013 1.0051
psi_z[2]   1566 1.0037
rho_z      1257 1.0048

Samples were drawn using NUTS(diag_e) at Wed Nov 24 17:43:37 2021.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

The fixed-effect estimates, in which our interest centers, are very similar to those for the model with only random slopes, reported in the text (cf., Table 25.5 on page 63). As was the case with mixed-effects models estimated by ML or REML, estimates of fixed effects are rarely sensitive to details of the specification of the random effects.

## References

J. Fox and S. Weisberg. *An R Companion to Applied Regression.* Sage, Thousand Oaks CA, third edition, 2019.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.