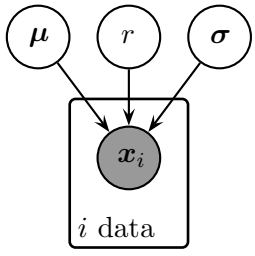
5 Some examples of data analysis

5.1 Pearson correlation

The Pearson product-moment correlation coefficient, usually denoted r, is a widely used measure of the relationship between two variables. It ranges from -1, indicating a perfect negative linear relationship, to $+1$, indicating a perfect positive relationship. A value of 0 indicates that there is no linear relationship. Usually the correlation r is reported as a single point estimate, perhaps together with a frequentist significance test.[1]

But, rather than just having a single number to measure the correlation, it would be nice to have a posterior distribution for r, saying how likely each possible level of correlation was. There are frequentist confidence interval methods that try to do this, as well as various analytic Bayesian results based on asymptotic approximations (e.g., Donner & Wells, 1986). An advantage of using a computational approach is the flexibility in the assumptions that can be made. It is possible to set up a graphical model that allows inferences about the correlation coefficient for any set of prior assumptions about the correlation.



$$
\begin{aligned}
\mu_1, \mu_2 &\sim \text{Gaussian}(0, 0.001) \\
\sigma_1, \sigma_2 &\sim \text{InvSqrtGamma}(0.001, 0.001) \\
r &\sim \text{Uniform}(-1, 1) \\
\boldsymbol{x}_i &\sim \text{MvGaussian}\left((\mu_1, \mu_2), \begin{bmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}^{-1} \right)
\end{aligned}
$$

Fig. 5.1 Graphical model for inferring a correlation coefficient.

One graphical model for doing this is shown in Figure 5.1. The observed data take the form $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ for the ith observation, and, following the theory behind the correlation coefficient, are modeled as draws from a multivariate Gaussian distribution. The parameters of this distribution are the means $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and standard deviations $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$ of the two variables, and the correlation coefficient r that links them.

[1] Frequentist or orthodox statistics is familiar to all cognitive scientists. Key frequentist concepts include the p-value, power, confidence intervals, and Type-I error rate. We believe that for scientific inference, the frequentist approach is inefficient at best and misleading at worst.

Box 5.1 Frequentist subjectivity

"Today one wonders how it is possible that orthodox logic continues to be taught in some places year after year and praised as 'objective', while Bayesians are charged with 'subjectivity'. Orthodoxians, preoccupied with fantasies about nonexistent data sets and, in principle, unobservable limiting frequencies—while ignoring relevant prior information—are in no position to charge anybody with 'subjectivity'." (Jaynes, 2003, p. 550).

In Figure 5.1, the standard deviations are assigned relatively uninformative inverse-square-root-gamma distributions. This is equivalent to placing gamma distributions on precisions, as was done in the seven scientists example in Section 4.2. The correlation coefficient itself is given a uniform prior over its possible range. All of these choices would be easily modified, with one obvious possible change being to give the prior for the correlation more density around 0.

The script `Correlation_1.txt` implements the graphical model in WinBUGS:

```
# Pearson Correlation
model{
  # Data
  for (i in 1:n){
    x[i,1:2] ~ dmnorm(mu[],TI[,])
  }
  # Priors
  mu[1] ~ dnorm(0,.001)
  mu[2] ~ dnorm(0,.001)
  lambda[1] ~ dgamma(.001,.001)
  lambda[2] ~ dgamma(.001,.001)
  r ~ dunif(-1,1)
  # Reparameterization
  sigma[1] <- 1/sqrt(lambda[1])
  sigma[2] <- 1/sqrt(lambda[2])
  T[1,1] <- 1/lambda[1]
  T[1,2] <- r*sigma[1]*sigma[2]
  T[2,1] <- r*sigma[1]*sigma[2]
  T[2,2] <- 1/lambda[2]
  TI[1:2,1:2] <- inverse(T[1:2,1:2])
}
```

The code `Correlation_1.m` or `Correlation_1.R` includes two data sets. Both involve fabricated data comparing response times in a semantic verification task (e.g., "Is a whale a fish?") on the x-axis with IQ measures on the y-axis, looking for a correlation between simple measures of decision-making and general intelligence.

For the first data set in the Matlab and R code, the results shown in Figure 5.2 are produced. The left panel shows a scatter-plot of the raw data. The right panel shows the posterior distribution of r, together with the standard frequentist point estimate.

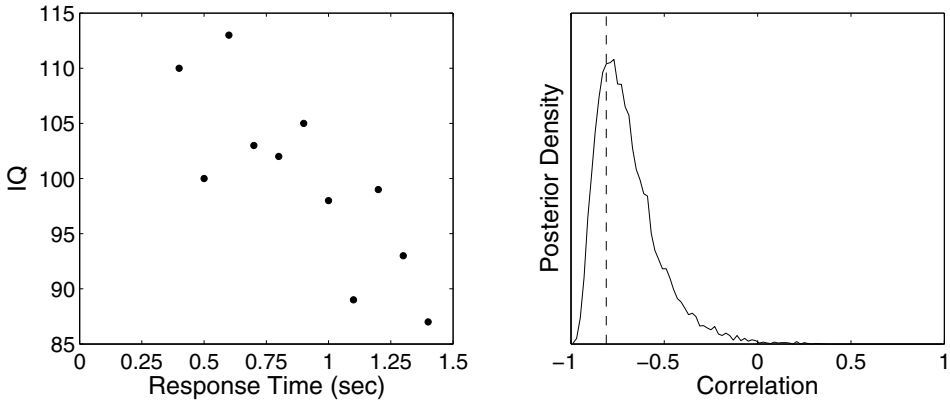


Fig. 5.2 Data (left panel) and posterior distribution for correlation coefficient (right panel). The broken line shows the frequentist point estimate.

Exercises

Exercise 5.1.1 The second data set in the Matlab and R code is just the first data set from Figure 5.2 repeated twice. Set `dataset=2` to consider these repeated data, and interpret the differences in the posterior distributions for r.

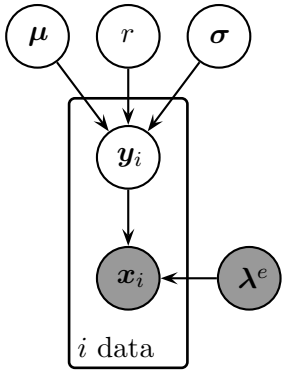
Exercise 5.1.2 Do you find the priors on μ_1 and μ_2 to be reasonable?

Exercise 5.1.3 The current graphical model assumes that the values from the two variables—the $\boldsymbol{x}_i = (x_{i1}, x_{i2})$—are observed with perfect accuracy. When might this be a problematic assumption? How could the current approach be extended to make more realistic assumptions?

5.2 Pearson correlation with uncertainty

We now tackle the problem asked by the last question in the previous section, and consider the correlations when there is uncertainty about the exact values of variables. It is likely that each individual response time is measured very accurately, since it is a physical quantity and good measurement tools exist. But the measurement of IQ seems likely to be less precise, since it is a psychological quantity, and measurement tools like IQ tests are less accurate. The uncertainty in measurement should be incorporated in an assessment of the correlation between the variables (e.g., Behseta, Berdyyeva, Olson, & Kass, 2009).

A simple approach for including this uncertainty is adopted by the graphical model in Figure 5.3. The observed data still take the form $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ for the ith person's response time and IQ measure. But these observations are now sampled from a Gaussian distribution, centered on the unobserved true response time and IQ of that person, denoted $\boldsymbol{y}_i = (y_{i1}, y_{i2})$. These true values are then modeled as the



$$\mu_1, \mu_2 \sim \text{Gaussian}(0, 0.001)$$
$$\sigma_1, \sigma_2 \sim \text{InvSqrtGamma}(0.001, 0.001)$$
$$r \sim \text{Uniform}(-1, 1)$$
$$\boldsymbol{y}_i \sim \text{MvGaussian}\left((\mu_1, \mu_2), \begin{bmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}^{-1}\right)$$
$$x_{ij} \sim \text{Gaussian}(y_{ij}, \lambda_j^e)$$

Fig. 5.3 Graphical model for inferring a correlation coefficient, when there is uncertainty inherent in the measurements.

$\boldsymbol{x}$ were in the previous model in Figure 5.1, as draws from a multivariate Gaussian distribution.

The precision of the measurements is captured by $\boldsymbol{\lambda}^e = (\lambda_1^e, \lambda_2^e)$ of the Gaussian draws for the observed data, $x_{ij} \sim \text{Gaussian}(y_{ij}, \lambda_j^e)$. The graphical model in Figure 5.3 assumes that these precisions are known.

The script `Correlation_2.txt` implements the graphical model shown in WinBUGS:

```
# Pearson Correlation With Uncertainty in Measurement
model{
  # Data
  for (i in 1:n){
    y[i,1:2] ~ dmnorm(mu[],TI[,])
    for (j in 1:2){
      x[i,j] ~ dnorm(y[i,j],lambdaerror[j])
    }
  }
  # Priors
  mu[1] ~ dnorm(0,.001)
  mu[2] ~ dnorm(0,.001)
  lambda[1] ~ dgamma(.001,.001)
  lambda[2] ~ dgamma(.001,.001)
  r ~ dunif(-1,1)
  # Reparameterization
  sigma[1] <- 1/sqrt(lambda[1])
  sigma[2] <- 1/sqrt(lambda[2])
  T[1,1] <- 1/lambda[1]
  T[1,2] <- r*sigma[1]*sigma[2]
  T[2,1] <- r*sigma[1]*sigma[2]
  T[2,2] <- 1/lambda[2]
  TI[1:2,1:2] <- inverse(T[1:2,1:2])
}
```

The code `Correlation_2.m` or `Correlation_2.R` uses the same data as in the previous section, but has different analyses because of the different assumptions about the uncertainty in measurement. In these new analyses, we assume that

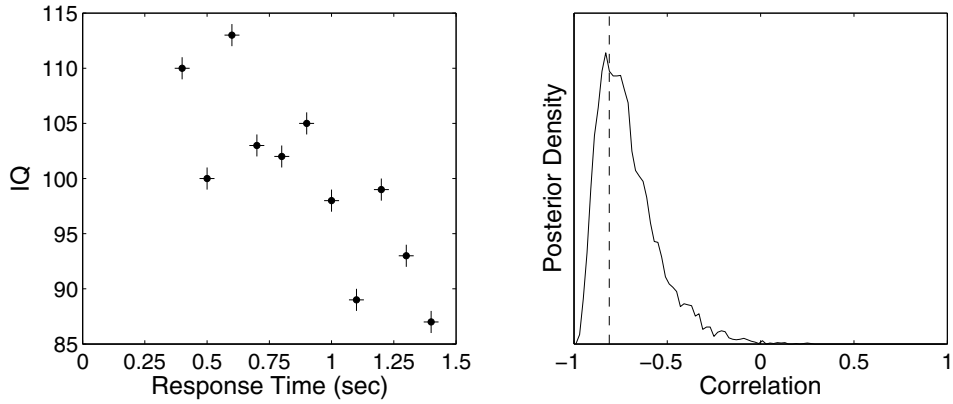


Fig. 5.4 Data (left panel), including error bars showing uncertainty in measurement, and posterior distribution for the correlation coefficient (right panel). The broken line shows the frequentist point estimate.

measurement uncertainty is originally expressed in terms of standard deviations, and then re-parameterized and supplied to the graphical model as precisions. The specific assumption is that $\sigma_1^e = .03$ for response times (which seem likely to be measured accurately) and $\sigma_2^e = 1$ for IQ (which seems near the smallest plausible value, so we assume that IQ is also measured accurately). The results of these assumptions using the model are shown in Figure 5.4. The left panel shows a scatterplot of the raw data, together with error bars representing the uncertainty quantified by the assumed standard deviations σ_1^e and σ_2^e. The right panel shows the posterior distribution of r, together with the standard frequentist point estimate.

Exercises

Exercise 5.2.1 Compare the results obtained in Figure 5.4 with those obtained earlier using the same data, in Figure 5.2, for the model without any account of uncertainty in measurement.

Exercise 5.2.2 Generate results for the second data set, which changes $\sigma_2^e = 10$ for the IQ measurement. Compare these results with those obtained assuming $\sigma_2^e = 1$.

Exercise 5.2.3 The graphical model in Figure 5.3 assumes the uncertainty for each variable is known. How could this assumption be relaxed to the case where the uncertainty is unknown?

Exercise 5.2.4 The graphical model in Figure 5.3 assumes the uncertainty for each variable is the same for all observations. How could this assumption be relaxed to the case where, for example, extreme IQs are less accurately measured than IQs in the middle of the standard distribution?

5.3 The kappa coefficient of agreement

An important statistical inference problem in a range of physical, biological, behavioral, and social sciences is to decide how well one decision-making method agrees with another. An interesting special case considers only binary decisions, and views one of the decision-making methods as giving objectively true decisions to which the other aspires. This problem occurs often in medicine, when cheap or easily administered methods for diagnosis are evaluated in terms of how well they agree with a more expensive or complicated "gold standard" method.

For this problem, when both decision-making methods make n independent assessments, the data $\boldsymbol{y}$ take the form of four counts: a observations where both methods decide "one," b observations where the objective method decides "one" but the surrogate method decides "zero," c observations where the objective method decides "zero" but the surrogate method decides "one," and d observations where both methods decide "zero," with $n = a + b + c + d$.

A variety of orthodox statistical measures have been proposed for assessing agreement using these data (but see Basu, Banerjee, & Sen, 2000; Broemeling, 2009, for Bayesian approaches). Useful reviews are provided by Agresti (1992), Banerjee, Capozzoli, McSweeney, and Sinha (1999), Fleiss, Levin, and Paik (2003), Kraemer (1992), Kraemer, Periyakoil, and Noda (2004), and Shrout (1998). Of all the measures, however, it is reasonable to argue that the conclusion of Uebersax (1987) that "the kappa coefficient is generally regarded as the statistic of choice for measuring agreement" (p. 140) remains true.

Cohen's (1960) kappa statistic estimates the level of observed agreement

$$p_o = \frac{a+d}{n}$$

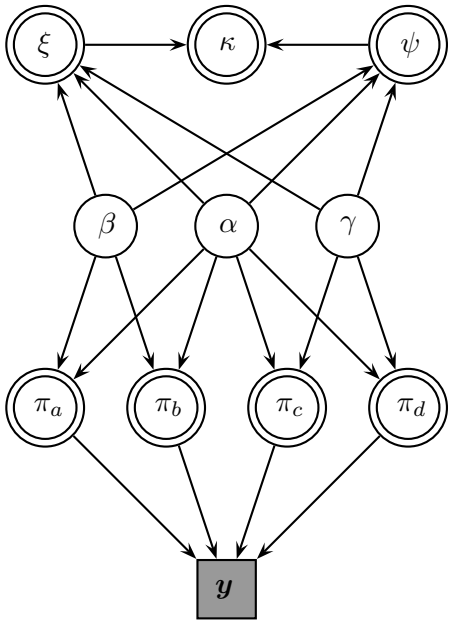
relative to the agreement that would be expected by chance alone (i.e., the overall probability for the first method to decide "one" times the overall probability for the second method to decide "one," and added to this the overall probability for the second method to decide "zero" times the overall probability for the first method to decide "zero")

$$p_e = \frac{(a+b)(a+c)+(b+d)(c+d)}{n^2},$$

and is given by

$$\kappa = \frac{p_o - p_e}{1 - p_e}.$$

Kappa lies on a scale of -1 to $+1$, with values below 0.4 often interpreted as "poor" agreement beyond chance, values between 0.4 and 0.75 interpreted as "fair to good" agreement beyond chance, and values above 0.75 interpreted as "excellent" agreement beyond chance (Landis & Koch, 1977). The key insight of kappa as a measure of agreement is its correction for chance agreement.



$$
\begin{aligned}
\kappa &\leftarrow (\xi - \psi) / (1 - \psi) \\
\xi &\leftarrow \alpha\beta + (1 - \alpha)\,\gamma \\
\psi &\leftarrow (\pi_a + \pi_b)\,(\pi_a + \pi_c) \\
&\quad + (\pi_b + \pi_d)\,(\pi_c + \pi_d) \\
\alpha, \beta, \gamma &\sim \text{Beta}(1, 1) \\
\pi_a &\leftarrow \alpha\beta \\
\pi_b &\leftarrow \alpha\,(1 - \beta) \\
\pi_c &\leftarrow (1 - \alpha)\,(1 - \gamma) \\
\pi_d &\leftarrow (1 - \alpha)\,\gamma \\
\boldsymbol{y} &\sim \text{Multinomial}([\pi_a, \pi_b, \pi_c, \pi_d]\,, n)
\end{aligned}
$$

Fig. 5.5 Graphical model for inferring the kappa coefficient of agreement.

The graphical model for a Bayesian version of kappa is shown in Figure 5.5. The key latent variables are α, β, and γ. The rate α is the rate at which the gold standard method decides "one." This means $(1 - \alpha)$ is the rate at which the gold standard method decides "zero." The rate β is the rate at which the surrogate method decides "one" when the gold standard also decides "one." The rate γ is the rate at which the surrogate method decides "zero" when the gold standard decides "zero." The best way to interpret β and γ is that they are the rate of agreement of the surrogate method with the gold standard, for the "one" and "zero" decisions respectively.

Using the rates α, β, and γ, it is possible to calculate the probabilities that both methods will decide "one," $\pi_a = \alpha\beta$, that the gold standard will decide "one" but the surrogate will decide "zero," $\pi_b = \alpha\,(1 - \beta)$, the gold standard will decide "zero" but the surrogate will decide "one," $\pi_c = (1 - \alpha)\,(1 - \gamma)$, and that both methods will decide "zero," $\pi_d = (1 - \alpha)\,\gamma$.

These probabilities, in turn, describe how the observed data, $\boldsymbol{y}$, made up of the counts a, b, c, and d, are generated. They come from a multinomial distribution with n trials, where on each trial there is a π_a probability of generating an a count, π_b probability for a b count, and so on.

So, observing the data $\boldsymbol{y}$ allows inferences to be made about the key rates α, β, and γ. The remaining variables in the graphical model in Figure 5.5 just re-express these rates in the way needed to provide an analogue to the kappa measure of chance-corrected agreement. The ξ variable measures the rate of agreement, which

is $\xi = \alpha\beta + (1-\alpha)\gamma$. The ψ variable measures the rate of agreement that would occur by chance, which is $\psi = (\pi_a + \pi_b)(\pi_a + \pi_c) + (\pi_b + \pi_d)(\pi_c + \pi_d)$, and could be expressed in terms of α, β, and γ. Finally κ is the chance-corrected measure of agreement on the -1 to $+1$ scale, given by $\kappa = (\xi - \psi)/(1 - \psi)$.

The script `Kappa.txt` implements the graphical model in WinBUGS:

```
# Kappa Coefficient of Agreement
model{
  # Underlying Rates
  # Rate Objective Method Decides "one"
  alpha ~ dbeta(1,1)
  # Rate Surrogate Method Decides "one" When Objective Method Decides "one"
  beta ~ dbeta(1,1)
  # Rate Surrogate Method Decides "zero" When Objective Method Decides "zero"
  gamma ~ dbeta(1,1)
  # Probabilities For Each Count
  pi[1] <- alpha*beta
  pi[2] <- alpha*(1-beta)
  pi[3] <- (1-alpha)*(1-gamma)
  pi[4] <- (1-alpha)*gamma
  # Count Data
  y[1:4] ~ dmulti(pi[],n)
  # Derived Measures
  # Rate Surrogate Method Agrees With the Objective Method
  xi <- alpha*beta+(1-alpha)*gamma
  # Rate of Chance Agreement
  psi <- (pi[1]+pi[2])*(pi[1]+pi[3])+(pi[2]+pi[4])*(pi[3]+pi[4])
  # Chance-Corrected Agreement
  kappa <- (xi-psi)/(1-psi)
}
```

The code `Kappa.m` or `Kappa.R` includes several data sets, described in the exercises below, for WinBUGS to sample from the graphical model.

Exercises

Exercise 5.3.1 *Influenza Clinical Trial.* Poehling, Griffin, and Dittus (2002) reported data evaluating a rapid bedside test for influenza using a sample of 233 children hospitalized with fever or respiratory symptoms. Of the 18 children known to have influenza, the surrogate method identified 14 and missed 4. Of the 215 children known not to have influenza, the surrogate method correctly rejected 210 but falsely identified 5. These data correspond to $a = 14$, $b = 4$, $c = 5$, and $d = 210$. Examine the posterior distributions of the interesting variables, and reach a scientific conclusion. That is, pretend you are a consultant for the clinical trial. What would your two- or three-sentence "take home message" conclusion be to your customers?

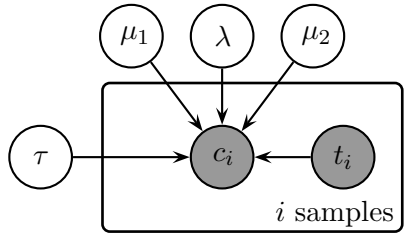
Exercise 5.3.2 *Hearing Loss Assessment Trial.* Grant (1974) reported data from a screening of a pre-school population intended to assess the adequacy of a school nurse assessment in relation to expert assessment of hearing loss. Of those children assessed by the expert as having hearing loss, 20 were correctly identified by the nurse and 7 were missed. Of those assessed by the expert

as not having hearing loss, 417 were correctly diagnosed by the nurse but 103 were incorrectly diagnosed as having hearing loss. These data correspond to $a = 20$, $b = 7$, $c = 103$, $d = 417$. Once again, examine the posterior distributions of the interesting variables, and reach a scientific conclusion. Once again, what would your two- or three-sentence "take home message" conclusion be to your customers?

Exercise 5.3.3 *Rare Disease.* Suppose you are testing a cheap instrument for detecting a rare medical condition. After 170 patients have been screened, the test results show that 157 did not have the condition, but 13 did. The expensive ground-truth assessment subsequently revealed that, in fact, none of the patients had the condition. These data correspond to $a = 0$, $b = 0$, $c = 13$, $d = 157$. Apply the kappa graphical model to these data, and reach a conclusion about the usefulness of the cheap instrument. What is special about this data set, and what does it demonstrate about the Bayesian approach?

5.4 Change detection in time series data

This case study involves near-infrared spectrographic data, in the form of oxygenated hemoglobin counts of frontal lobe activity during an attention task in Attention Deficit Hyperactivity Disorder (ADHD) adults. The interesting modeling problem is that a change is expected in the time series of counts because of the attention task. The statistical problem is to identify the change. To do this, we are going to make a number of strong assumptions. In particular, we will assume that the counts come from a Gaussian distribution that always has the same variance, but changes its mean at one specific point in time. The main interest is therefore in making an inference about this change point.



$$\mu_1, \mu_2 \sim \text{Gaussian}(0, 0.001)$$
$$\lambda \sim \text{Gamma}(0.001, 0.001)$$
$$\tau \sim \text{Uniform}(0, t_{\max})$$
$$c_i \sim \begin{cases} \text{Gaussian}(\mu_1, \lambda) & \text{if } t_i < \tau \\ \text{Gaussian}(\mu_2, \lambda) & \text{if } t_i \geq \tau \end{cases}$$

Fig. 5.6 Graphical model for detecting a single change point in time series.

Figure 5.6 presents a graphical model for detecting the change point. The observed data are the counts c_i at time t_i for the ith sample. The unobserved variable τ is the time at which the change happens, which controls whether the counts have mean μ_1 or μ_2. A uniform prior over the full range of possible times is assumed for

the change point, and generic weakly informative priors are given to the means and the precision.

The script `ChangeDetection.txt` implements this graphical model in WinBUGS:

```
# Change Detection
model{
  # Data Come From A Gaussian
  for (i in 1:n){
    c[i] ~ dnorm(mu[z1[i]],lambda)
  }
  # Group Means
  mu[1] ~ dnorm(0,.001)
  mu[2] ~ dnorm(0,.001)
  # Common Precision
  lambda  ~ dgamma(.001,.001)
  sigma  <- 1/sqrt(lambda)
  # Which Side is Time of Change Point?
  for (i in 1:n){
    z[i]  <- step(t[i]-tau)
    z1[i] <- z[i]+1
  }
  # Prior On Change Point
  tau ~ dunif(0,n)
}
```

Note the use of the `step` function. This function returns 1 if its argument is greater than or equal to zero, and 0 otherwise. The `z1` variable, however, serves as an indicator variable for `mu`, and therefore it needs to take on values 1 and 2. This is the reason `z` is transformed to `z1`. Study this code and make sure you understand what the `step` function accomplishes in this example.

The code `ChangeDetection.m` or `ChangeDetection.R` applies the model to the near-infrared spectrographic data. Uniform sampling is assumed, so that $t = 1, \ldots, 1178$.

The code produces a simple analysis, finding the mean of the posteriors for τ, μ_1 and μ_2, and using these summary points to overlay the inferences over the raw data. The result looks something like Figure 5.7. The time series data themselves are shown by the jagged black lines. The expected value of the posterior mean for the pre- and post-change levels, given by the posterior means for μ_1 and μ_2, are shown by the horizontal lines. The expected change point, given by the posterior mean for τ, is just under 800 samples, and is used to separate the plotting of the pre-change level from the post-change level.

Exercises

Exercise 5.4.1 Draw the posterior distributions for the change point, the means, and the common standard deviation.

Exercise 5.4.2 Figure 5.7 shows the mean of the posterior distribution for the change point (this is the point in time where the two horizontal lines meet). Can you think of a situation in which such a plotting procedure can be misleading?

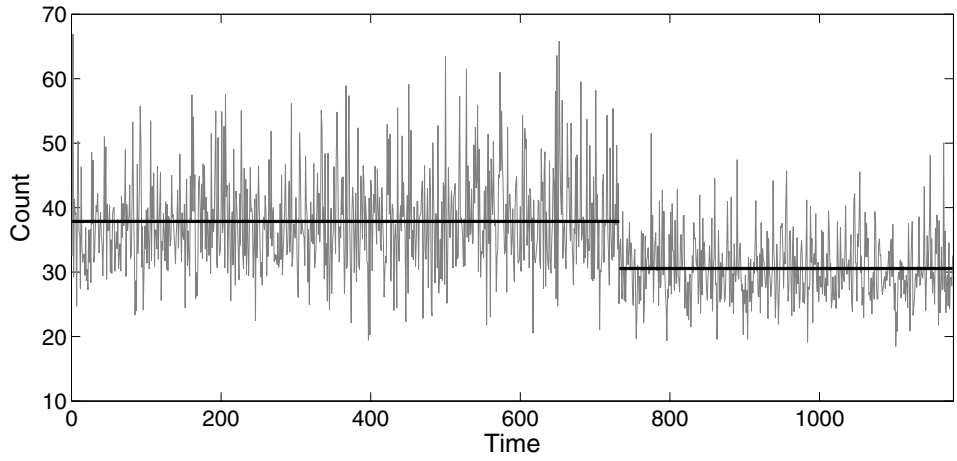


Fig. 5.7 Identification of a change point in time series data. The time series are shown by the jagged lines (note that these are observed data to be modeled; they are not chains from MCMC sampling), and the pre- and post-change levels around the expected change point are shown by the two overlaid horizontal lines.

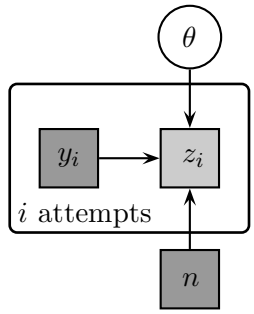
Exercise 5.4.3 Imagine that you apply this model to a data set that has two change points instead of one. What could happen?

5.5 Censored data

Starting April 13 2005, Cha Sa-soon, a 68-year-old grandmother living in Jeonju, South Korea, repeatedly tried to pass the written exam for a driving license. In South Korea, this exam features 50 four-choice questions. In order to pass, one requires a score of at least 60 points out of a maximum of 100. Accordingly, we assume that each correct answer is worth two points, so that in order to pass, one needs to answer at least 30 questions correctly.

What has made Cha Sa-soon something of a national celebrity is that she failed to pass the test on 949 consecutive occasions, spending the equivalent of 4200 US dollars on application fees. In her last, 950th attempt, Cha Sa-soon scored the required minimum of 30 correct questions and finally passed her written exam. After her 775th failure, in February 2009, Mrs Cha told Reuters news agency, "I believe you can achieve your goal if you persistently pursue it. So don't give up your dream, like me. Be strong and do your best."

We know that on her final and 950th attempt, Cha Sa-soon answered 30 questions correctly. In addition, news agencies report that in her 949 unsuccessful attempts, the number of correct answers had ranged from 15 to 25. Armed with this knowledge, what can we say about θ, the latent probability that Cha Sa-soon can answer



$$\theta \sim \text{Uniform}(0.25, 1)$$
$$z_i \sim \text{Binomial}(\theta, n)$$
$$15 \leq z_i \leq 25 \quad \text{if } y_i = 1$$

Fig. 5.8 Graphical model for inferring a rate from observed and censored data.

any one question correctly? Note that we assume each question is equally difficult, and that Cha Sa-soon does not learn from her earlier attempts.

The Cha Sa-soon data are special because we do not know the precise scores for the failed attempts. We only know that these scores range from 15 to 25. In statistical terms, these data are said to be censored, both from below and from above. We follow an approach inspired by Gelman and Hill (2007, p. 405) to apply WinBUGS to the problem of dealing with censored data.

Figure 5.8 presents a graphical model for dealing with the censored data. The variable z_i represents both the first 949 unobserved, and the final observed attempt. This means z_i is observed once, but not observed the other times. This sort of variable is known as *partially observed*, and is denoted in the graphical model by a lighter shading, between the dark shading of fully observed nodes, and the lack of shading for fully unobserved or latent nodes.

The variable y_i is a simple binary indicator variable, denoting whether or not the ith attempt is observed. The bounds $z^{\text{lo}} = 15$ and $z^{\text{hi}} = 25$ give the known censored interval for the unobserved attempts. Finally, $n = 50$ is the number of questions in the test. This means that $z_i \sim \text{Binomial}(\theta, n)_{\mathcal{I}(z^{\text{lo}}, z^{\text{hi}})}$ when y_i indicates a censored attempt, but that z_i is not censored for the final known score $z_{950} = 30$. The probability of a correct answer to a question, θ, is given a uniform prior between 0.25 and 1, corresponding to the assumption that chance accuracy of 1 in 4 is the lowest possible probability.

The script `ChaSaSoon.txt` implements this graphical model in WinBUGS:

```
# ChaSaSoon Censored Data
model{
  for (i in 1:nattempts){
    # If the Data Were Unobserved y[i]=1, Otherwise y[i]=0
    z.low[i]  <- 15*equals(y[i],1)+0*equals(y[i],0)
    z.high[i] <- 25*equals(y[i],1)+n*equals(y[i],0)
    z[i] ~ dbin(theta,n)I(z.low[i],z.high[i])
  }
  # Uniform Prior on Rate Theta
  theta ~ dbeta(1,1)I(.25,1)
}
```

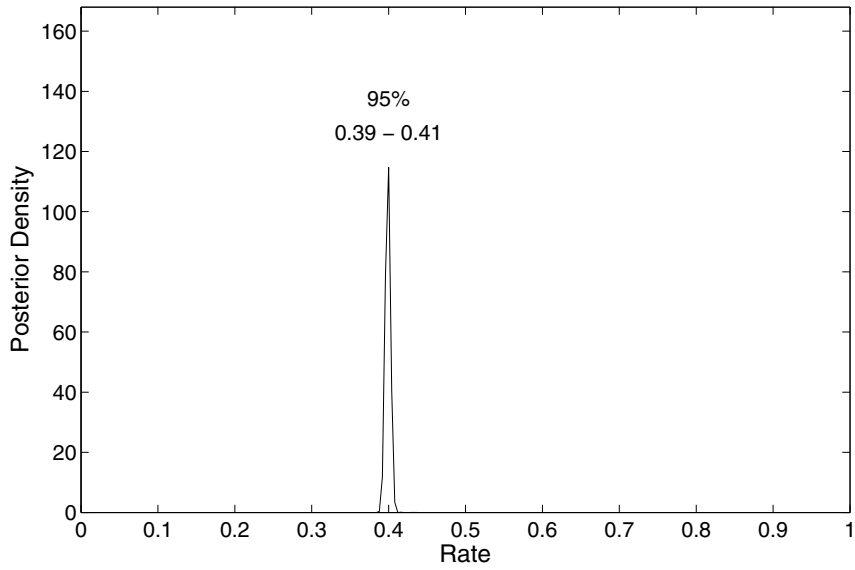


Fig. 5.9 Posterior density for Cha Sa-soon's rate of answering questions correctly.

Note the use of the `equals` command, which returns 1 when its arguments match, and 0 when they mismatch. Thus, when `y[i]=1`, for censored data, `z.low[i]` is set to 15 and `z.hi[i]` is set to 25. When `y[i]=0`, `z.low[i]` is set to 0 and `z.hi[i]` is set to n. These `z.low[i]` and `z.hi[i]` values are then applied to censor the binomial distribution that generates the test scores, using the WinBUGS `I` ("interval") command. In this way, the use of `equals` implements what might be considered the "case" or "if-then-else" logic of the model.

The code `ChaSaSoon.m` or `ChaSaSoon.R` applies the model to the data from Cha Sa-soon.[2] The posterior density for θ is shown in Figure 5.9, and can be seen to be relatively peaked. Despite the fact that we do not know the actual scores for 949 of the 950 results, we are still able to infer a lot about θ.

Exercises

Exercise 5.5.1 Do you think Cha Sa-soon could have passed the test by just guessing?

Exercise 5.5.2 What happens when you increase the interval in which you know the data are located, from 15–25 to something else?

Exercise 5.5.3 What happens when you decrease the number of failed attempts?

Exercise 5.5.4 What happens when you increase Cha Sa-soon's final score from 30?

[2] On some computers, WinBUGS will persistently return the mysterious error message "value of binomial `z[950]` must be greater than lower bound." If you know how to fix this error, we would love to hear from you. Otherwise, we can only suggest you run the code on a different computer.

Exercise 5.5.5 Do you think the assumption that all of the scores follow a binomial distribution with a single rate of success is a good model for these data?

5.6 Recapturing planes

An interesting inference problem that occurs in a number of fields is to estimate the size of a population, when a census is impossible, but repeated surveying is possible. For example, the goal might be to estimate the number of animals in a large woodland area that cannot be searched exhaustively. Or, the goal might be to decide how many students are on a campus, but it is not possible to count them all. Or, the goal might be to find out how many words in a given language a person knows, but it is not feasible to ask the person to list them all.

A clever sampling approach to this problem is given by capture-and-recapture methods. The basic idea is to capture (i.e., identify, tag, or otherwise remember) a sample at one time point, and then collect another sample. The number of items in the second sample that were also in the first then provides relevant information as to the population size. High recapture counts suggest that the population is small, and low recapture counts suggest that the population is large.

Probably the simplest possible version of this approach can be formalized with t as the unknown population size, x as the size of the first sample (i.e., number of units captured), and n as the size of the second sample from which a subset of k units were also present in the first sample (i.e., number of units recaptured). That is, first x animals are tagged or people remembered or words produced, then k out of n are seen again when a second sample is taken.

The statistical model to relate the counts and make inferences about the population size t is based on the hypergeometric distribution. The probability of seeing k items recaptured in a sample of size n, from the x originally captured in a population of size t, is

$$\Pr\left(K = k\right) = \binom{x}{k}\binom{t-x}{n-k} \Big/ \binom{t}{n}.$$

Intuitively, the second sample involves taking n items from a population of t, and has k out of x recaptures, and $n-k$ other items out of the other $t-x$ in the population. Another way to formalize this is to say that the number of recaptures k is a sample from a hypergeometric distribution

$$k \sim \text{Hypergeometric}\big(n, x, t\big).$$

To make these ideas concrete, consider the challenge of estimating how many aircraft a small airline company has in its fleet. One day at an airport, you see 10 of the airline company's planes parked at adjacent gates, and record their unique identifying tail numbers. A few days later, at a different airport, you see 5 of the same company's planes. Looking at the tail number of those planes, you observe

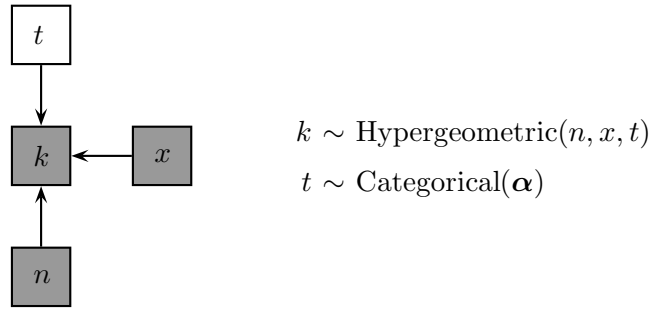


Fig. 5.10 Graphical model for inferring a population from capture-and-recapture data.

that 4 of the 5 were part of your original list. This is a capture-and-recapture problem with $x = 10$, $k = 4$, and $n = 5$.

The Bayesian approach to this problem involves assigning a prior to t, and using the hypergeometric distribution as the appropriate likelihood function. Conceptually, this means $k \sim \text{Hypergeometric}(n, x, t)$, as in the graphical model in Figure 5.10. The vector $\boldsymbol{\alpha}$ allows for any sort of prior mass to be given to all the possible counts for the population total. Since $x+(n-k)$ items are known to exist, one reasonable choice of prior might be to make every possibility from $x+(n-k)$ to $t^{\max}$ equally likely, where $t^{\max}$ is a sensible upper bound on the possible population. Suppose, for example, in the airplane problem that you know that the maximum number the company could possibly have is 50 planes, so that $t^{\max} = 50$.

While it is simple conceptually, there is a difficulty in implementing the graphical model in Figure 5.10. The problem is that WinBUGS does not provide the hypergeometric distribution. It is, however, possible to implement distributions that are not provided, but for which the likelihood function can be expressed in WinBUGS. This can be done using either the so-called "ones trick" or the "zeros trick."[3] These tricks rely on simple properties of the Poisson and Bernoulli distributions. By implementing the likelihood function of the new distribution within the Poisson or Bernoulli distribution, and forcing values of 1 or 0 to be sampled, it can be shown that the samples actually generated will come from the desired distribution.

The script `Planes.txt` implements the graphical model in Figure 5.10 in WinBUGS, using the zeros trick. Note how the terms in the log-likelihood expression for the hypergeometric distribution are built up to define `phi`, and a constant `C` is used to ensure the Poisson distribution is used with a positive value:

```
# Planes
model{
  # Hypergeometric Likelihood Via Zeros Trick
  logterm1 <- logfact(x)-logfact(k)-logfact(x-k)
  logterm2 <- logfact(t-x)-logfact(n-k)-logfact((t-x)-(n-k))
  logterm3 <- logfact(t)-logfact(n)-logfact(t-n)
  C <- 1000
```

[3] Using the zeros trick or ones trick in JAGS involves putting the assignment of `zeros` or `ones` inside the data definition block, rather than inside the model definition block.

Box 5.2 The zeros trick, ones trick, and WBDev

The zeros trick and ones trick are extremely useful, and relatively easy to implement in many cases, but a little difficult to understand conceptually. The key insight is that the negative log-likelihood of a sample of 0 from $\text{Poisson}(\phi)$ is ϕ, and similarly for a sample of 1 from $\text{Bernoulli}(\theta)$ it is θ. So, by setting $\log \phi$ or θ appropriately, and forcing 1 or 0 to be observed, sampling effectively proceeds from the distribution defined by ϕ or θ.

More complicated extensions to the distributions and functions available in WinBUGS require using the WinBUGS Development Interface (WBDev: Lunn, 2003). This is an add-on program that allows the user to hand-code functions and distributions in Component Pascal. Wetzels, Lee, and Wagenmakers (2010) provide a tutorial on WBDev that includes simple worked examples of defining new distributions and functions. More detailed cognitive science applications are provided by Wetzels, Vandekerckhove, et al. (2010) implementing the Expectancy-Valence model of decision-making as a function in WBDev, and Vandekerckhove et al. (2011) implementing the drift-diffusion model as a distribution in WBDev. Both of these applications would be impractical without WBDev.

```
  phi <- -(logterm1+logterm2-logterm3)+C
  zeros <- 0
  zeros ~ dpois(phi)
  # Prior on Population Size
  for (i in 1:tmax){
    tptmp[i] <- step(i-(x+n-k))
    tp[i] <- tptmp[i]/sum(tptmp[1:tmax])
  }
  t ~ dcat(tp[])
}
```

The code `Planes.m` or `Planes.R` applies the model to the data $x = 10$, $k = 4$, and $n = 5$, using uniform prior mass for all possible sizes between $x + (n - k) = 11$ and $t^{\max} = 50$. The posterior distribution for t is shown in Figure 5.11. The inference is that it is mostly likely there are not many more than 11 planes, which makes intuitive sense, since 4 out of 5 in the second sample were from the original set of 10.

Exercises

Exercise 5.6.1 Try changing the number of planes seen again in the second sample from $k = 4$ to $k = 0$. What inference do you draw about the population size now?

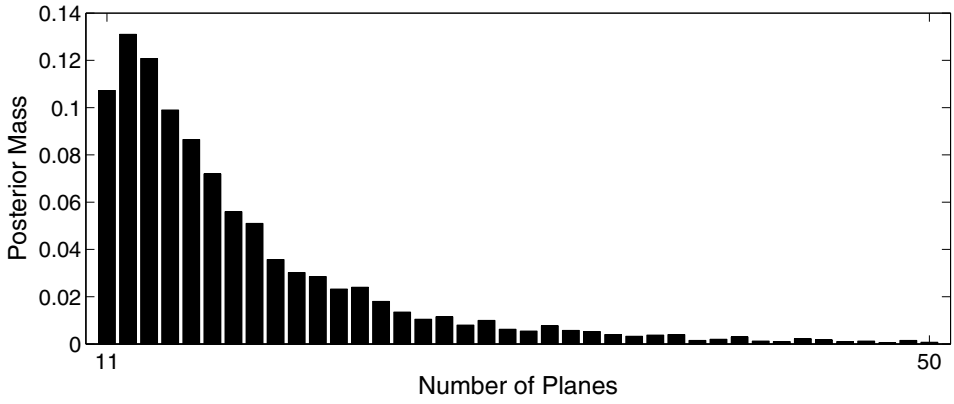


Fig. 5.11 Posterior mass for the number of planes, known to be 50 or fewer, based on a capture-recapture experiment with $x = 10$ planes in the first sample, and $k = 4$ out of $n = 5$ seen again in the second sample.

Exercise 5.6.2 How much impact does the upper bound $t^{\max} = 50$ have on the final conclusions when $k = 4$ and when $k = 0$? Develop your answer by trying both the $k = 4$ and $k = 0$ cases with $t^{\max} = 100$.

Exercise 5.6.3 Suppose, having obtained the posterior mass in Figure 5.11, the same fleet of planes was subjected to a new sighting at a different airport at a later day. What would be an appropriate prior for t?