23. Ordinal Predicted Variable

Ben Blayney

10/06/2020

Examples:

- Happiness ratings on 1-7 scale as a function of their total financial assets.
- Movie ratings as a function of the year they were made.

Traditional method: ordinal regression / ordered probit regression

Bayesian GLM approach:

- Inverse-link function is a thresholded *cumulative normal* (probit_.)
- Noise distribution is a categorical distribution

23.1 Modelling Ordinal Data with an Underlying Metric Variable

- Ordinal data is often a bracketing/bucketing of an underlying metric value (which may be normally distributed)
- Even if the underlying data is normally distributed the ordinal values may not resemble a normal distribution (Fig 23.1)
- The probability of a particular ordinal outcome is the area under the normal curve between the thresholds of that outcome.
  - Can be calculated using differences in the cumulative normal distribution: $p(y = k|\mu, \sigma, \{\theta_j\}) = \Phi((\theta_k - \mu)/\sigma) - \Phi((\theta_{k-1} - \mu)/\sigma)$ where θ_k are the thresholds of the outcomes.
- **Takeaway**: we might be able to model a distribution of ordinal values as consecutive intervals of a normal distribution on an underlying metric scale with appropriately partitioned thresholds.
  - **BB: this doesn't really help you in modelling if the thresholds can be chosen freely? If they cannot be chosen freely then we are applying some strange constraints and we usually would not really have much information on how to apply these or why they would be a good model of the data?**

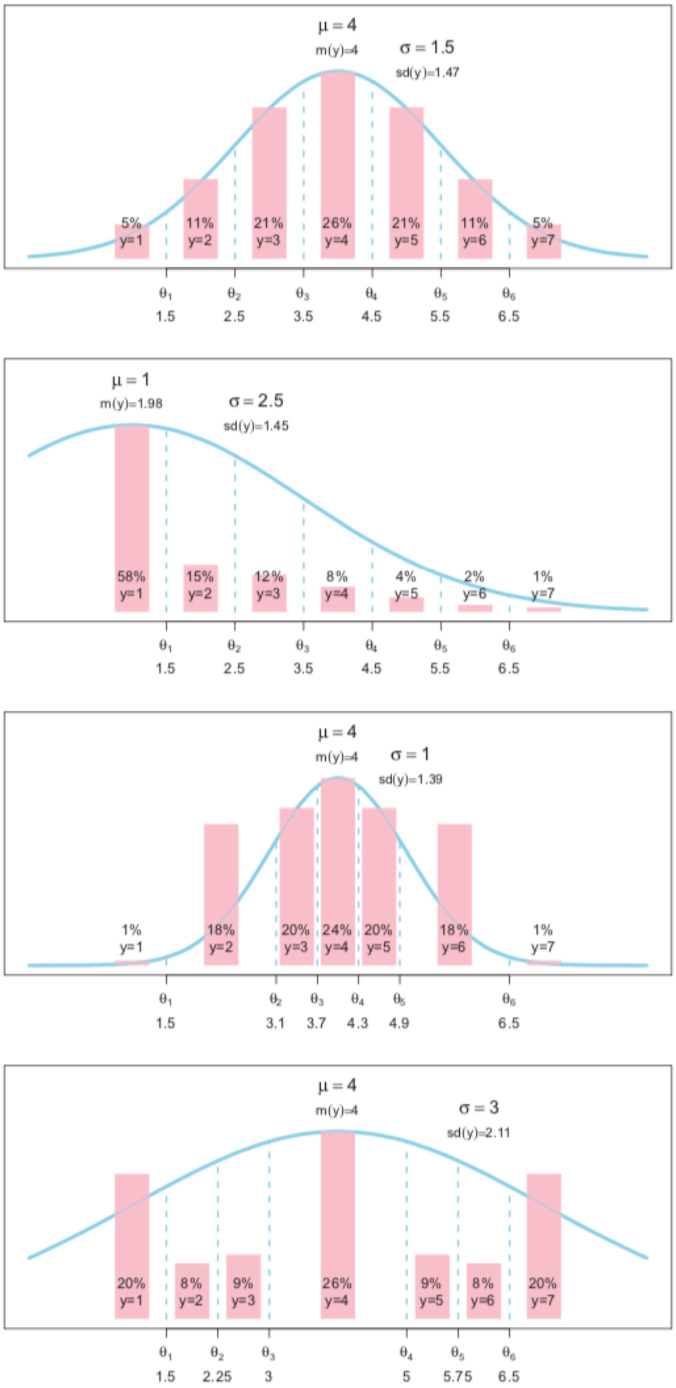


Figure 23.1: Examples of ordinal data generated by the thresholded cumulative-normal model. Ordinal outcome values (y) are indicated by bars. The horizontal axis is the underlying continuous value, which has a normal distribution with mean μ and standard deviation σ. The mean and standard deviation of the ordinal data values (treated as if metric) are annotated as m(y) and sd(y). Thresholds between ordinal outcome values on underlying continuous scale are denoted θ_j. See main text for remarks about each example. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

23.2 The Case of a Single Group

- If there are K ordinal values, the model has K+1 parameters: $\theta_1, ..., \theta_{K-1}, \mu, \sigma$.
  - If we increase all the thresholds by a that gives the same result as decreasing μ by a
  - Similar with expanding or contracting the thresholds and varying σ
  - No uniquely correct choice of how to "pin down" the axis by setting two of the parameters, we choose:
    - $\theta_1 = 1 + 0.5$
    - $\theta_{K-1} = K - 0.5$
    - *BB: end up with mean and standard deviation effectively being controlled by amount of data in the edge groups, then the middle groups fit themselves around what's left..?*

23.2.1 Implementaion in JAGS

```
"
model {
  for (i in 1:Ntotal) {
    y[i] ~ dcat(pr[i,1:nYlevels])
    pr[i,1] <- pnorm(thresh[1], mu, 1/sigma^2)
    for (k in 2:(nYlevels-1)) {
      pr[i,k] <- max(0, pnorm(thresh[k], mu, 1/sigma^2) - pnorm(thresh[k-1], mu, 1/sigma^2))
    }
    pr[i,nYlevels] <- 1 - pnorm(thresh[nYlevels-1], mu, 1/sigma^2)
  }
}
"
```

Notes:

- cumulative normal in JAGS is called pnorm
- max(0,. . .) needed because thresholds randomly generated by MCMC chain so possible to get nonsensical result without max

Priors *BB: again note uniform prior over log range; I think people should be more careful about this. . .*:

```
"
mu ~ dnorm((1_nYlevels)/2, 1/(nYlevels)^2)
sigma ~ dunif(nYlevels/1000, nYlevels*10)
for (k in 2:(nYlevels-2)) {
  thresh[k] ~ dnorm(k+0.5, 1/2^2)
}
"
```

23.2.2 Examples: Bayesian estimation recovers true parameter values

Figure 23.2 and 23.3. Goals is to show that Bayesian estimation can accurately recover our parameter values even when μ is extreme and the thresholds are not evenly spaced. Figure 23.2 data generated using $\mu = 1,\ \sigma = 2.5,\ \theta_1 = 1.5,\ \theta_2 = 2.5,\ \theta_3 = 3.5,\ \theta_4 = 4.5,\ \theta_5 = 5.5,\ \theta_6 = 6.5.$

Remember parameter values need to be *jointly* credible: if we have a higher θ_2 threshold and want to maintain the likelihood of outcome 3 then we need to also increase θ_3. Causes a domino effect through all the thresholds.

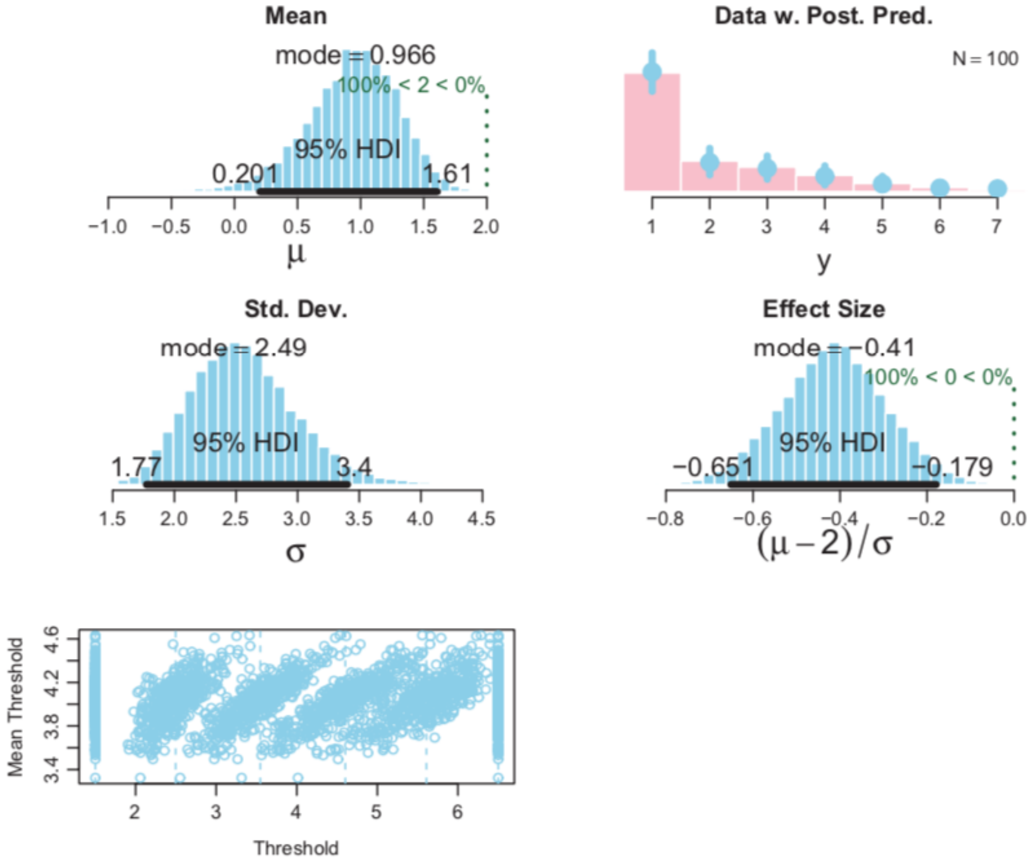


Figure 23.2: Bayesian posterior distribution for one group of ordinal data. True generating parameters are $\mu = 1.0$, $\sigma = 2.5$, $\theta_1 = 1.5$, $\theta_2 = 2.5$, $\theta_3 = 3.5$, $\theta_4 = 4.5$, $\theta_5 = 5.5$, and $\theta_6 = 6.5$. The Bayesian estimation accurately recovers the generating parameters. Posterior distribution clearly excludes a comparison value of $\mu = 2.0$. The posterior predictive distribution accurately describes the data distribution. *NHST treating data as metric:* The mean is *not* significantly different from a comparison value of $\mu = 2.0$: $M = 1.95$, $t = 0.36$, $p = 0.722$, with 95% CI of 1.67 to 2.23, and with effect size $d = 0.036$. The sample standard deviation is $S = 1.40$. The t test describes the data as normally distributed, which clearly is not the case here. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

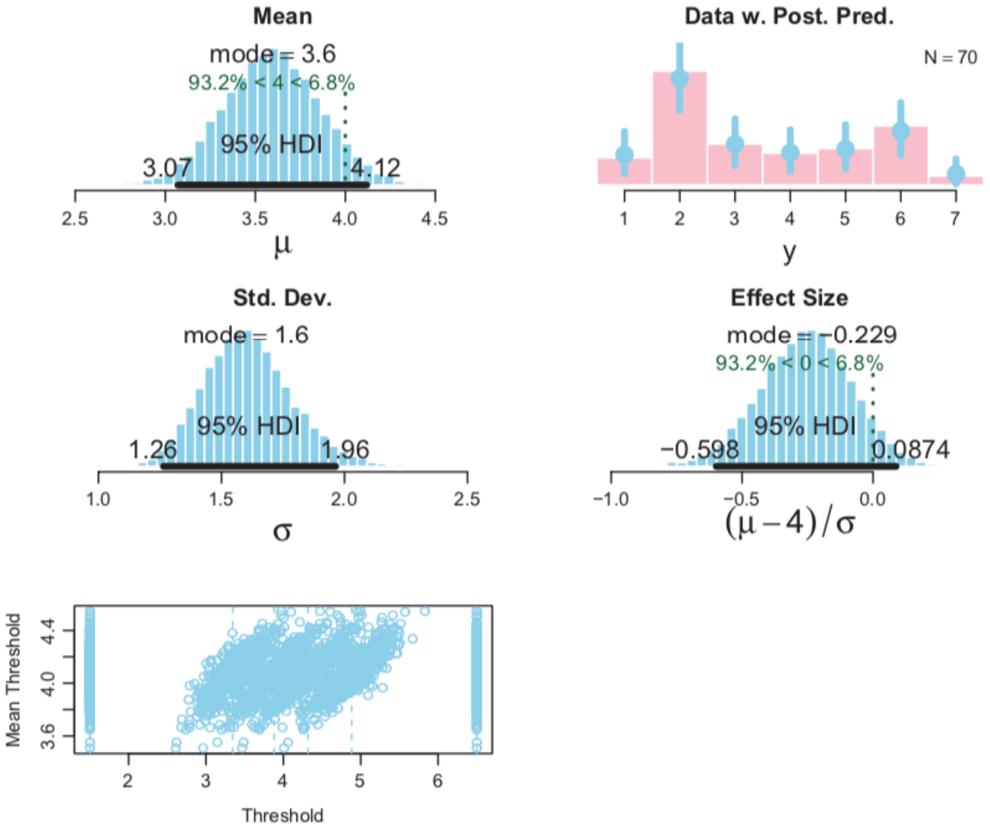


Figure 23.3: Bayesian posterior distribution for one group of ordinal data. True generating parameters are $\mu = 3.5$ and $\sigma = 1.5$, with $\theta_1 = 1.5$, $\theta_2 = 3.3$, $\theta_3 = 3.8$, $\theta_4 = 4.2$, $\theta_5 = 4.7$, and $\theta_6 = 6.5$. The Bayesian estimation accurately recovers the generating parameters. Posterior distribution includes (does not reject) a comparison value of $\mu = 4.0$. Importantly, the posterior predictive distribution nicely describes the data distribution. *NHST treating data as metric:* The mean is significantly different from $\mu = 4.0$: $M = 3.47$, $t = 2.47$, $p = 0.016$, with 95% CI of 3.04 to 3.90, and effect size $d = 0.295$. The sample standard deviation is $S = 1.79$. The t test describes the data as normally distributed which clearly is not the case here. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

23.2.2.1 Not the same results as pretending the data are metric

If instead we treat the data as though it were metric and normally distributed we get very different results:

- Mean of 1.95 instead of 0.966 (data generated using mean of 1.0).
- Sometimes the mean is ok but the predictions of individual outcome probabilities are terrible

BB: is it really that this special way of treating ordinal data is good and treating it as metric is bad, or is it really that when they were assuming the data was metric they assumed it was normally distributed when it clearly wasn't, how would other metric data models compare?

23.2.2.2 Ordinal outcomes versus Likert scales

Likert scales are from 1=strongly disagree to 5=strongly agree. Remember for this chapter we are interested in describing the ordinal outcomes themselves, not the arithmetic average of several ordinal responses. If there are many related items on a questionnaire then the model of the ordinal data can use latent factors to express relations among the items.

23.3 The Case of Two Groups

Example: ask about Happiness for two groups - one after watching sports car ads, one after watching charity ads. *Assume that both groups access the same underlying metric scale and thresholds, what differs is the mean and variance of the feeling.*

23.3.1 Implementation in JAGS

Simple extention of the specification for one group.

23.3.2 Examples: Not funny

Again showing that comparison of means and std deviations of likert values for two groups are much better made through this Bayesian method than a traditional method that assumes the data is normal.

23.4 The Case of Metric Predictors

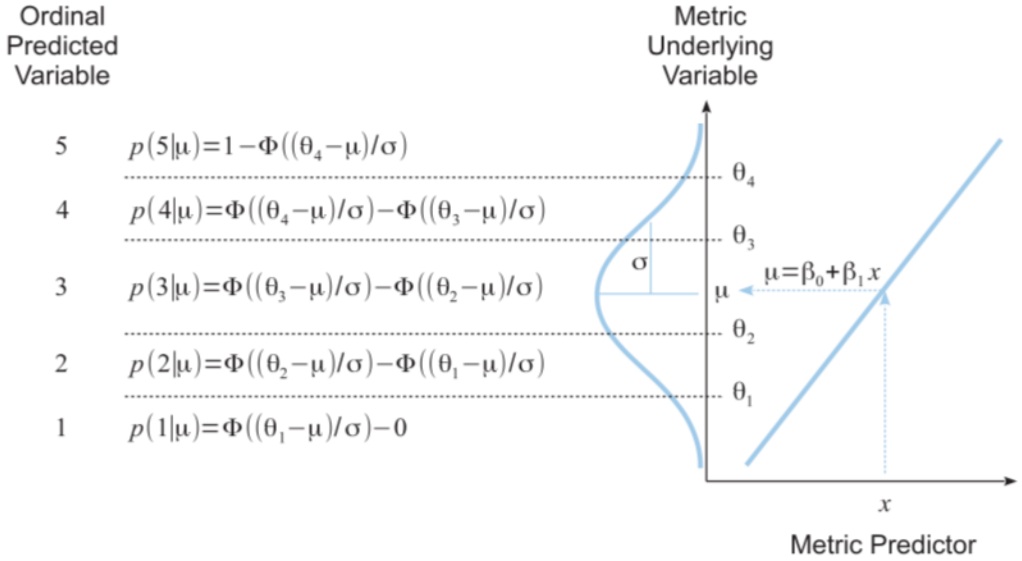

Ordinal Predicted Variable
Metric Underlying Variable
5
$p(5|\mu)=1-\Phi((\theta_4-\mu)/\sigma)$
4
$p(4|\mu)=\Phi((\theta_4-\mu)/\sigma)-\Phi((\theta_3-\mu)/\sigma)$
3
$p(3|\mu)=\Phi((\theta_3-\mu)/\sigma)-\Phi((\theta_2-\mu)/\sigma)$
2
$p(2|\mu)=\Phi((\theta_2-\mu)/\sigma)-\Phi((\theta_1-\mu)/\sigma)$
1
$p(1|\mu)=\Phi((\theta_1-\mu)/\sigma)-0$
θ_4
θ_3
σ
μ
$\mu=\beta_0+\beta_1 x$
θ_2
θ_1
x
Metric Predictor

Figure 23.6: Thresholded cumulative-normal regression. Right side shows metric predictor variable mapped to metric underlying variable, as in simple linear regression of Figure 17.1 (p. 462). Left side shows mapping from metric underlying to observed ordinal variable, displaying Equations 23.1–23.3 at the corresponding intervals between thresholds. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Figure 23.6 for important explanation of how we do linear regression to predict the underlying metric variable, then we apply our same model to convert underlying metric variable to ordinal predicted variable. Ordinal probit regression.

Figures 23.7 to 23.10 interesting examples.

23.4.1 Implementation in JAGS

Combination of linear regression models seen previously and the model we just illustrated above in this chapter. Predictors were standardized to improve MCMC efficiency, as we have also seen before. Predicted values are just categories and don't need to be standardized so it is more similar to logistic regression in a way.

23.4.2 Examples: Happiness and money

Data fall mostly in the middle of the response scale (Fig 23.7 and Fig 23.8) Underlying regression is linear but it should asymptote so this is not a good model of the underlying value: there should be diminishing returns.

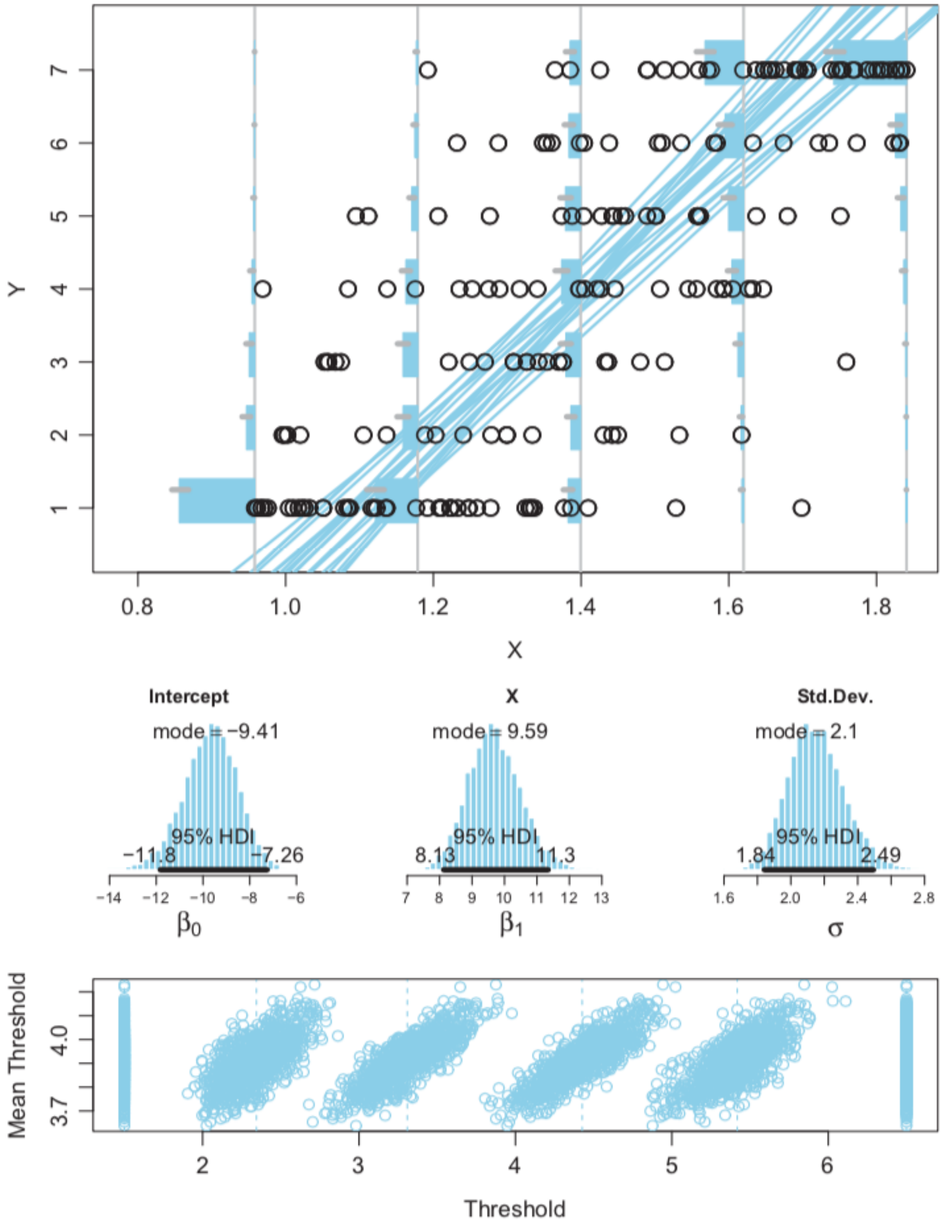


Figure 23.7: Upper panel: Data are shown as dots. Horizontal bars show mean posterior predicted probability at selected values of the predictor as marked by the vertical lines. Grey segments at tops (i.e., left end) of bars show the 95% HDI of posterior predicted probability. A smattering of credible regression lines is superimposed. The true generating parameters are $\beta_0 = -10.0$, $\beta_1 = 10.0$, $\sigma = 2.0$, $\theta_1 = 1.5$, $\theta_2 = 2.5$, $\theta_3 = 3.5$, $\theta_4 = 4.5$, $\theta_5 = 5.5$, $\theta_6 = 6.5$. Lower panels: The Bayesian estimation accurately recovers the generating parameters, as indicated by the marginal posterior distributions. *Least-squares estimate treating data as metric:* $\beta_0 = -5.42$ (SE=0.61), $\beta_1 = 6.71$ (SE=0.43), $\sigma = 1.52$. Notice that the slope is badly underestimated by least-squares estimation in this case. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

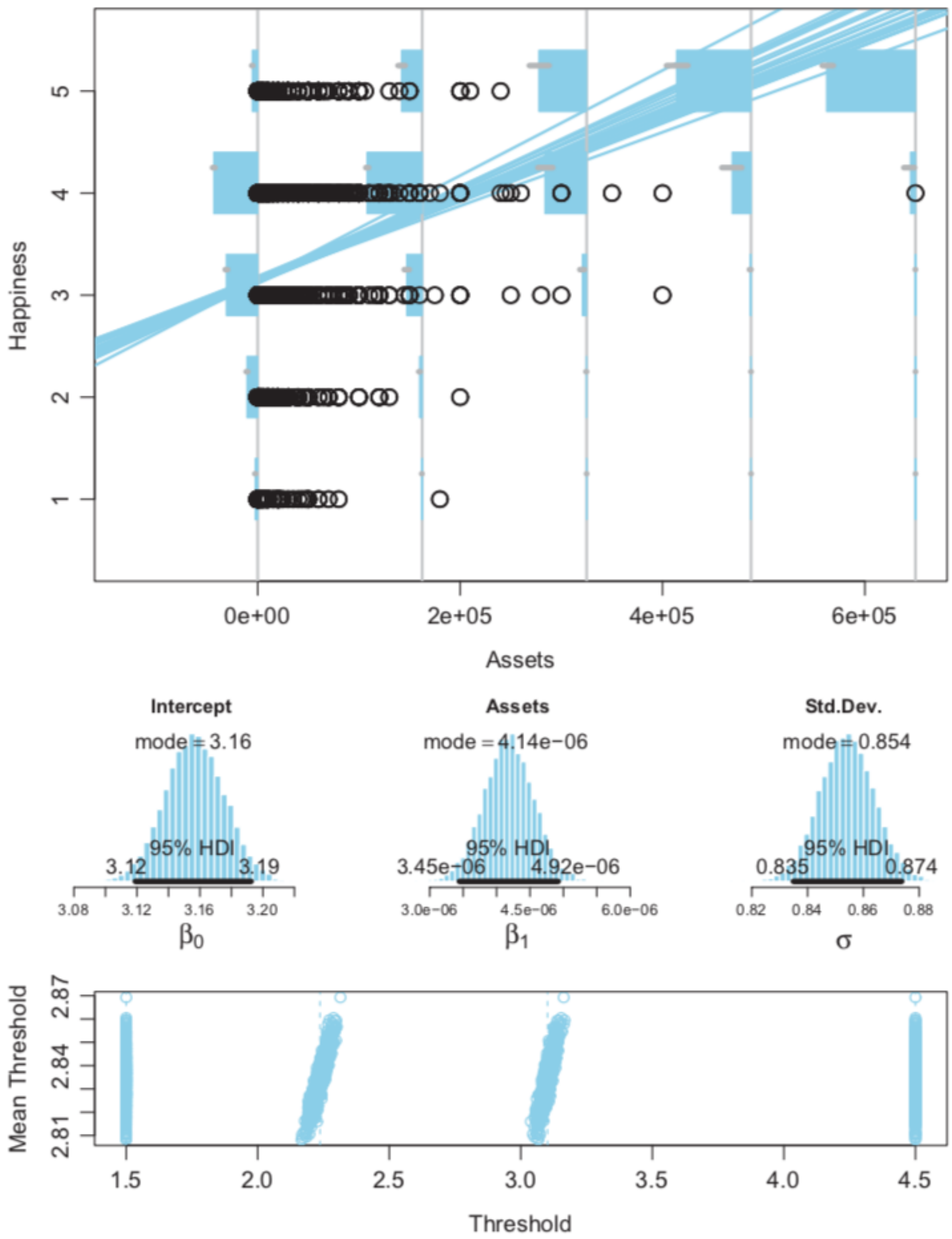


Figure 23.8: Upper panel: Happiness as a function of total household assets ($N = 6,835$, data from Shi, 2009). Assets are in 2002 Chinese yaun (2e+05 yuan was equivalent to about 24,200 US dollars in 2002). Horizontal bars show mean posterior predicted probability at selected values of the predictor as marked by the vertical lines. Grey segments at tops (i.e., left end) of bars show the 95% HDI of posterior predicted probability. A smattering of credible regression lines is superimposed. Lower panels show marginal posterior distribution on parameters. *Least-squares estimate treating data as metric:* $\beta_0 = 3.425$ (SE=0.012), $\beta_1 = 3.82e - 6$ (SE=3.39e-7), $\sigma = 0.847$. Notice that the slope is estimated to be smaller by least-squares estimation in this case, and predictive probabilities are different than Bayesian. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

23.4.3 Example: Movies - They don't make 'em like they used to

Have threshold lines that are level contours of the underlying metric planar surface (Fig 23.9)

However, the region the threshold lines bound as reserved for categories 7 and 1 don't have any datapoints in.

23.4.4 Why are some thresholds outside the data?

Compare to artificial data (Fig 23.10) as an example; has many more data points that span the entire range.

Fig 23.11 is good for showing outcomes for noisy data. Y axis is probability of each outcome (different line for each outcome). Most probable outcome within interval is not necessarily the outcome centered on that interval

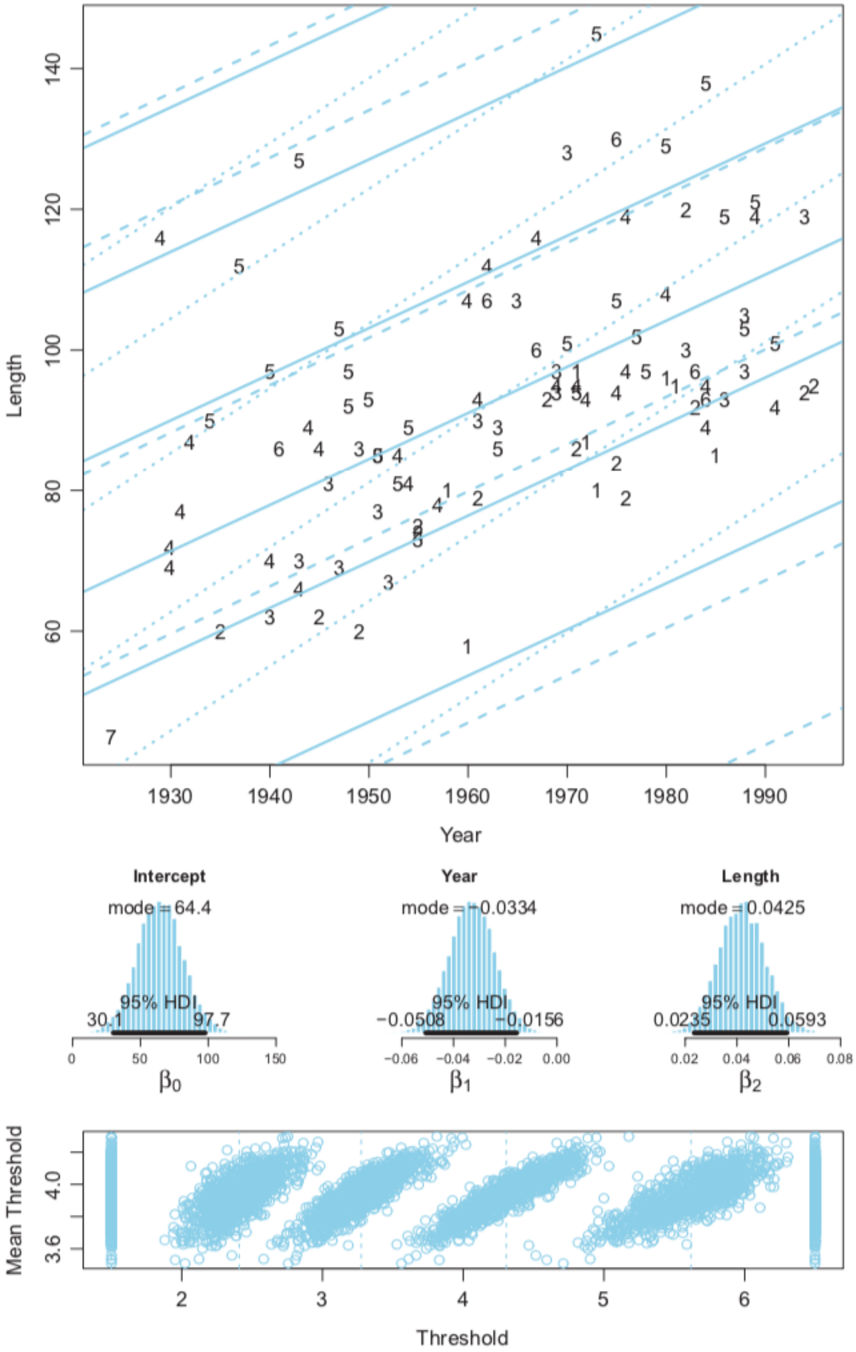


Figure 23.9: Analysis of movie rating data from Moore (2006). Not shown is the marginal posterior on σ, which has a modal value of about 1.25 and an 95% HDI from about 1.0 to 1.5. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

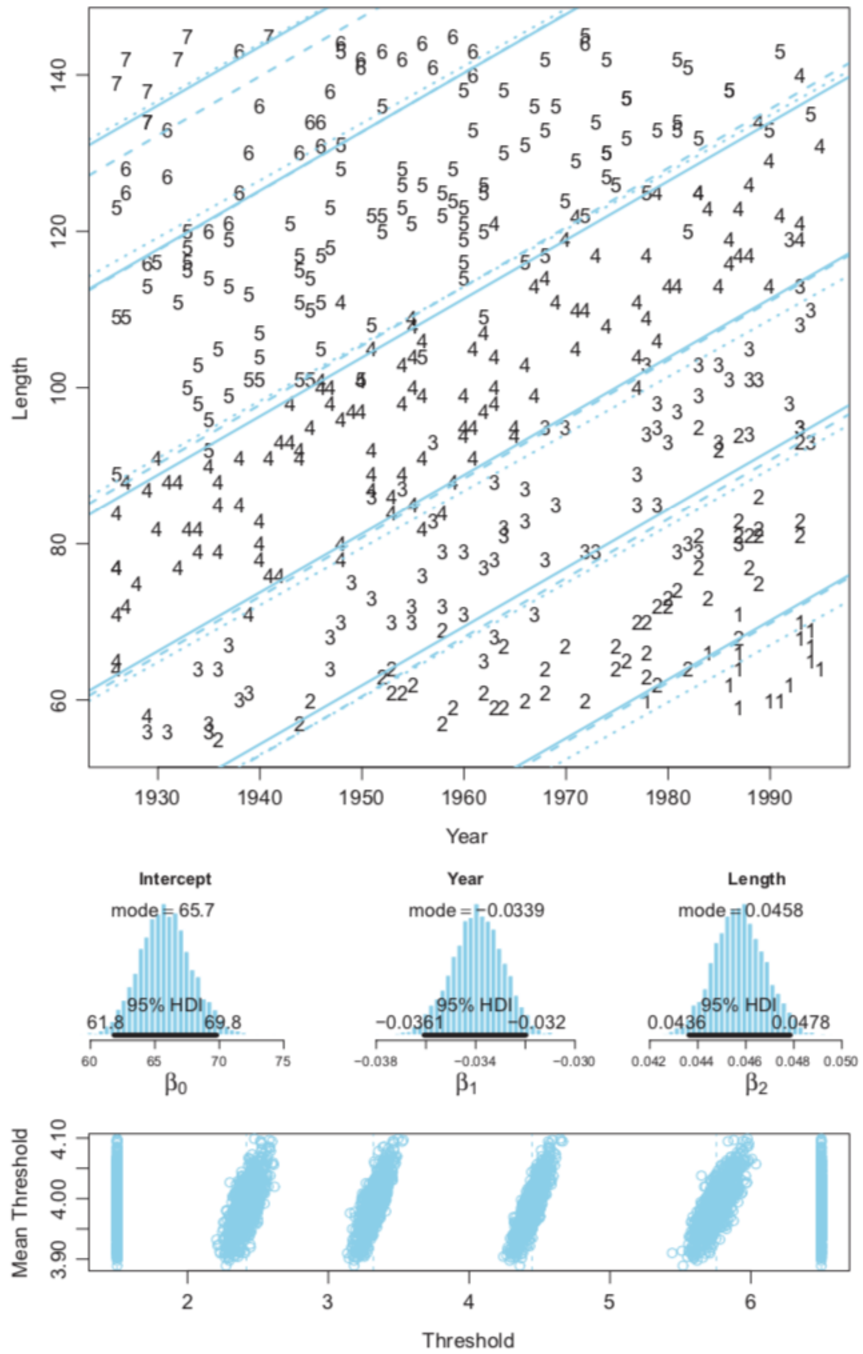


Figure 23.10: Analysis of simulated movie-rating data. These artificial data have very small noise, with generating $\sigma = 0.20$, compared with $\sigma \approx 1.25$ in the actual data of Figure 23.9. (The intercept and slope parameters are set a bit differently than the actual data so that all outcome values are present in the range of the predictors.) Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

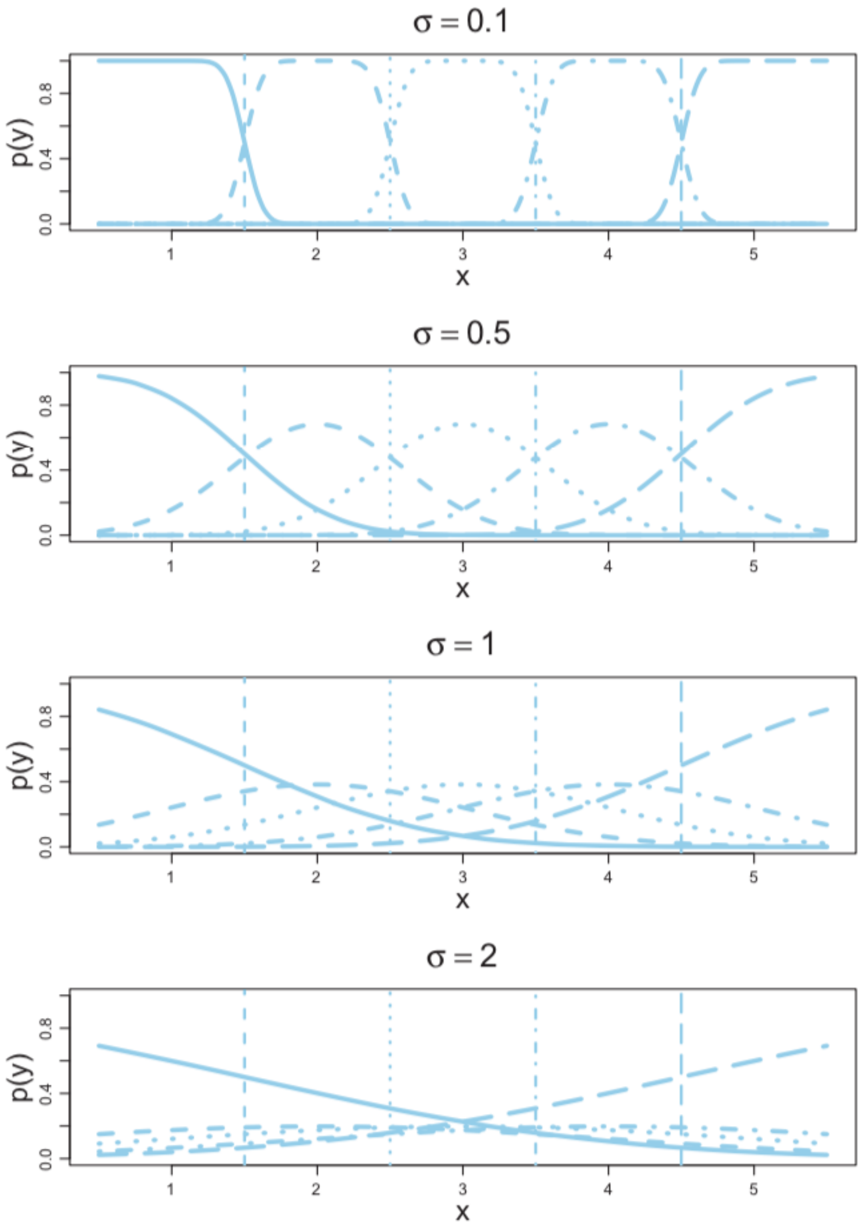


Figure 23.11: The probability of an ordinal response as a function of the predictor x. Thresholds are arbitrarily set at $\theta_1 = 1.5$, $\theta_2 = 2.5$, $\theta_3 = 3.5$, and $\theta_4 = 4.5$. Upper panel is for small noise ($\sigma = 0.1$), and lower panel is for large noise ($\sigma = 2.0$). Note that x values in the data might span only a small range on the x axis. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

23.5 Posterior Prediction

How are posterior predicted probabilities with 95% HDI computed? At each step in MCMC chain compute predicted outcome probabilities using equations given and explained. From the full chain of credible posterior predicted outcome probabilities the central tendency and 95% HDI are then computed. Brief example how to do this below: use the MCMC chain coda object returned from JAGS (codaSample) then do the processing of it in R.

```
mcmcMat = as.matrix(codaSamples)
chainLength = nrow(mcmcMat)
beta0 = mcmcMat[,"beta0"]
beta1 = mcmcMat[,"beta"]
thresh = mcmcMat[,grep("^thresh",colnames(mcmcMat))]
sigma = mcmcMat[,"sigma"]

outProb = matrix(NA,nrow=chainLength,ncol=max)
for (stepIdx in 1:chainLength) {
  mu = beta0[stepIdx] + beta1[stepIdx] * xProbe
  threshCumProb = pnorm((thres[stepIdx, ] - mu)/sigma[stepIdx])
  outProb[stepIdx, ] = c(threshCumProb,1) - c(0,threshCumProb)
}
outMed = apply(outProb, 2, median, na.rm=True)
outHdi = apply(outProb, 2, HDIofMCMC)
```

23.6 Generalizations and Extensions

Approaches to outliers:

- Use heavy-tailed distribution instead of normal distribution to describe noise. Can use cumulative t function which is build into JAGS as pt (analogous to pnorm).
- Model the data as being from two different sources as we did in Robust Logistic Regression; one source is our cumulative normal model and the other is "guessing"; so we mix our models predicted probabilities with some random guessing probability.

Variable selection: just as predictors in linear regression or logistic regression can be given inclusions parameters, so can our predictors in thresholded cumulative-normal regression.

Nominal predictors: works just fine, look back to Figure 21.12 to see how this might work. The only change is putting thresholds on the normal noise distribution to create probabilities for the ordinal outcomes.

23.7 Exercises

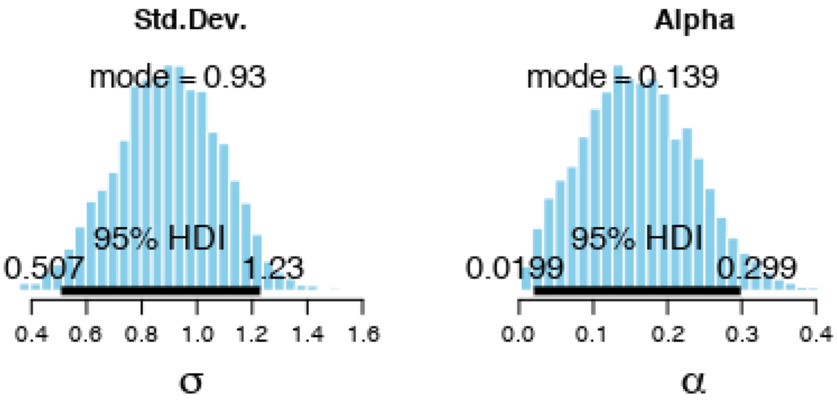
Exercise 23.2 [Purpose: Modifying the program to]

```
setwd("./DBDA2Eprograms")
source("DBDA2E-utilities.R") # Load definitions of graphics functions etc.
source("Jags-Yord-XmetMulti-Mnormal-Guessing-Example.R")
```

```
setwd("./DBDA2Eprograms")
source("DBDA2E-utilities.R") # Load definitions of graphics functions etc.
source("Jags-Yord-XmetMulti-Mnormal-tDist-Example.R")
```

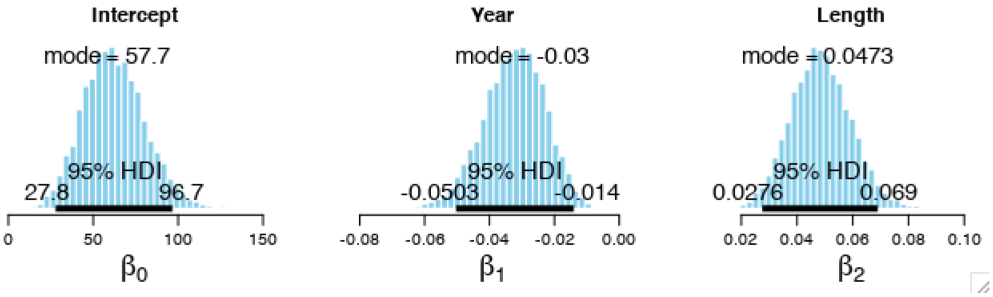
A

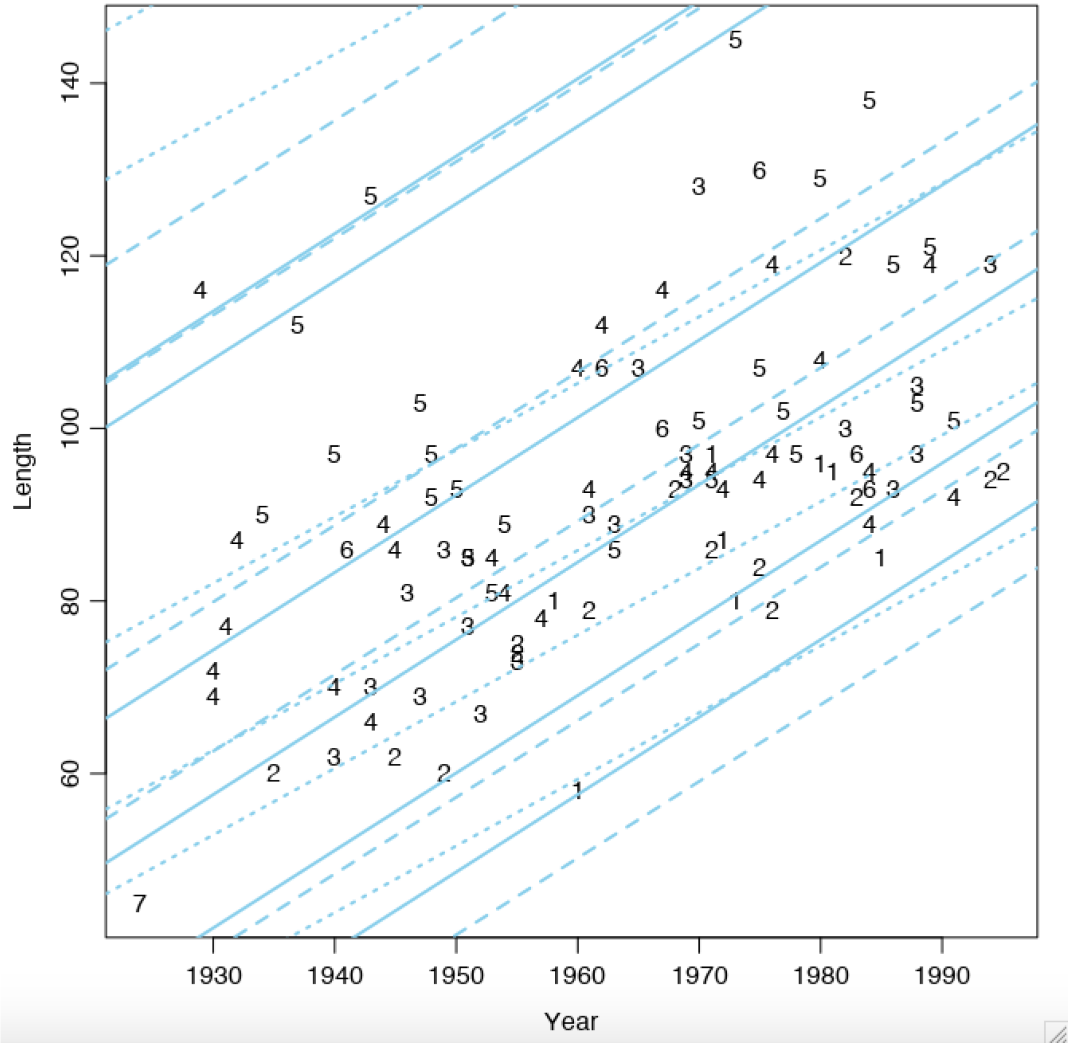
i.) Posterior distribution of the guessing parameter (alpha). Recall guessing parameter is $\frac{1}{7} = 0.1429$; this gets multiplied by alphas itself to get the probability added into each category.

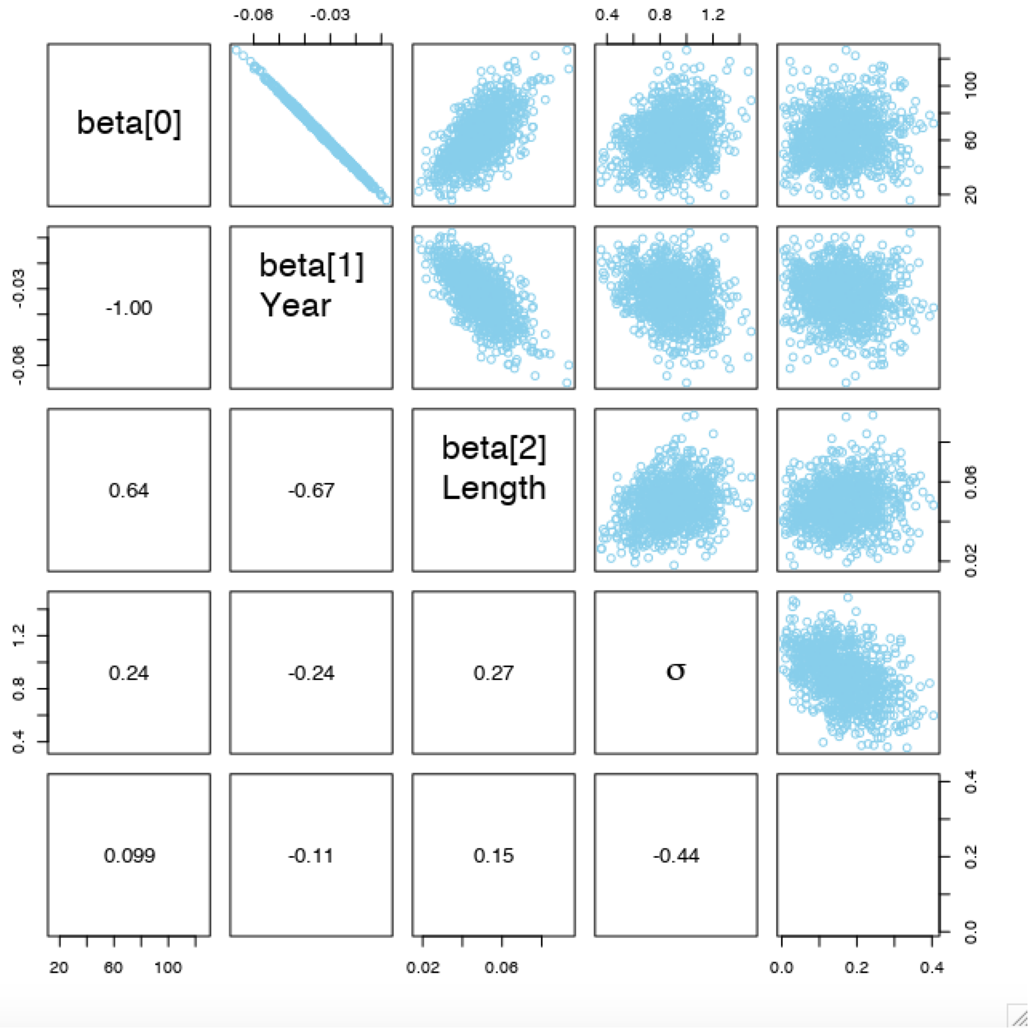


Interesting (and expected) sigma is definitely reduced compared to not using a guessing parameter.

ii.) Are the regression coefficients a little more extreme? Yes. **Why?**



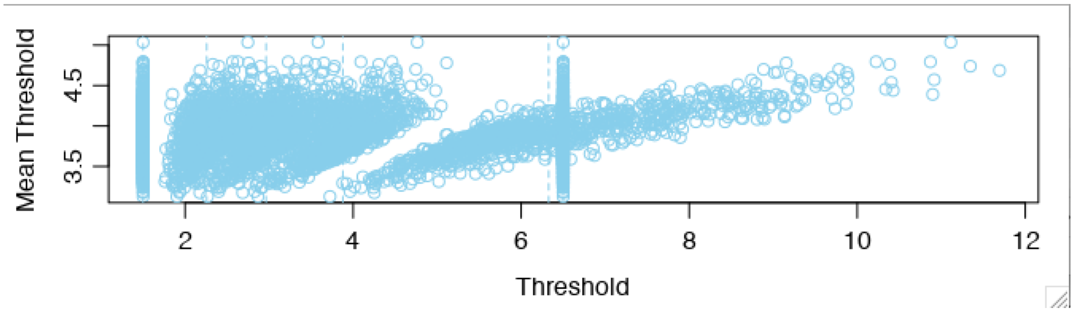

Length
140
120
100
80
60
1930
1940
1950
1960
1970
1980
1990
Year



iii.) Is there anything unusual about the posterior distribution on the thresholds (see Figure 23.12), and why? Hint: even if the thresholds are randomly inverted in the MCMC chain, and the outcome probabilities from that component are zero, the random component still gives the outcomes a nonzero probability. Thresholds can get inverted; thresholds are generated randomly by the MCMC chain so sometimes it's possible they become the wrong way around.

Normally (without random guessing mixture) when this happens we model that the probability of any data occuring in the category between the thresholds is 0. Therefore when we use the data to work out the likelihood and find there is some data in that category we get the likelihood of that to be 0, which indicates that parameters when the thresholds are the wrong way around are not credible, and this will be reflected in the posterior as well. Now the way MCMC Gibbs sampling works is that it spends time at particular parameters proportional to their posterior probabilities; if the posterior probabilities are 0 then those parameter combinations won't be explored and won't show up in the figure.

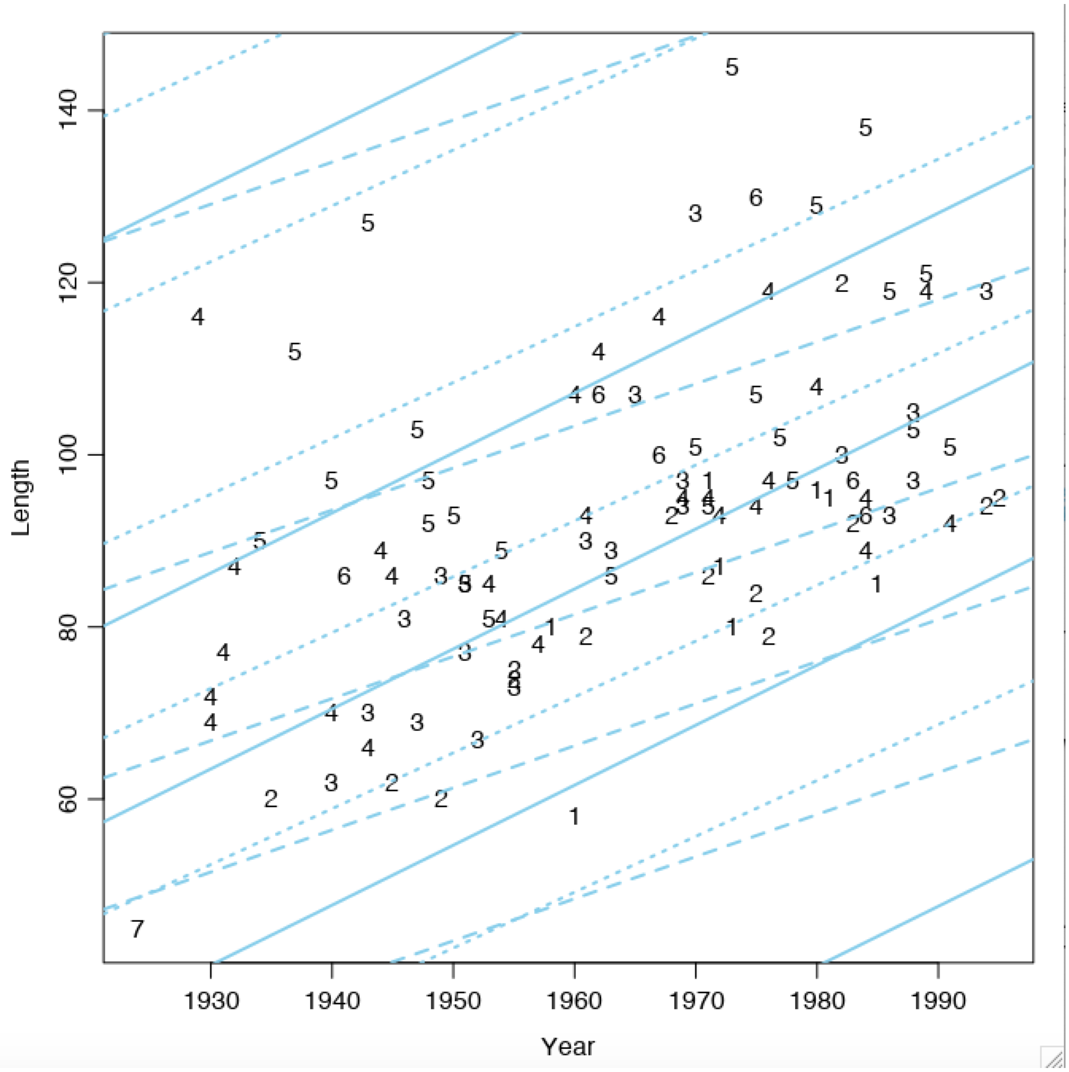
When we include the guessing mixture then the model now has some nonzero probability for data to be in any category no matter what the thresholds are, so suddenly the MCMC chain is now able to explore all these values.

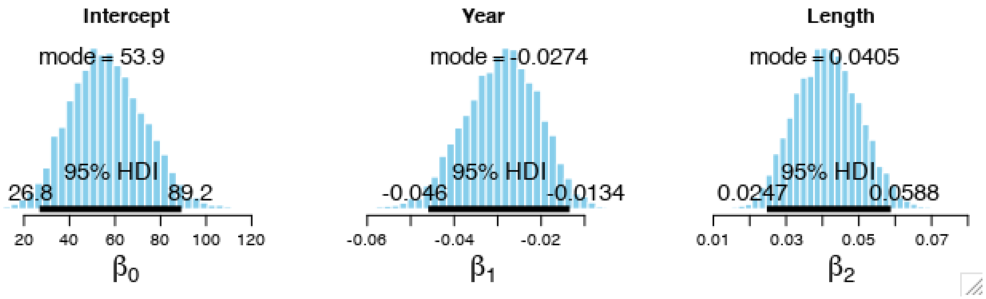


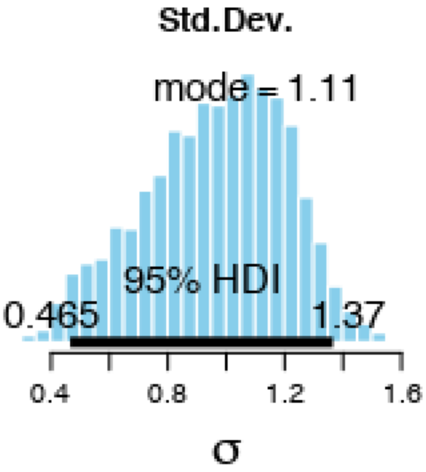
B

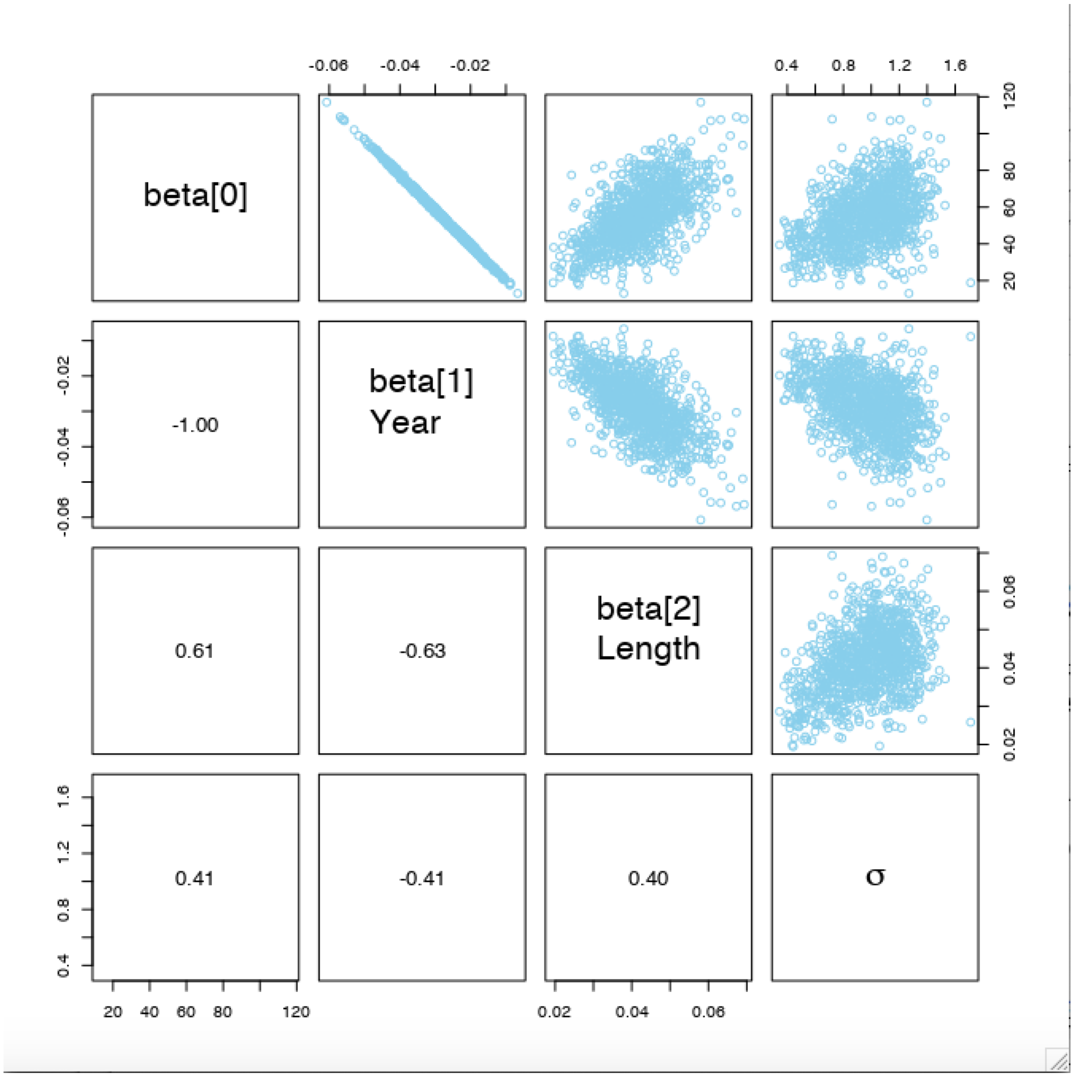
Now change model to a thresholded cumulative-t-distribution model (with no random guessing mixture)

i.) Are the regression coefficients a little more extreme? Seem a little less extreme, sigma is somewhere between part A and the original example where the modal value was about 1.25.

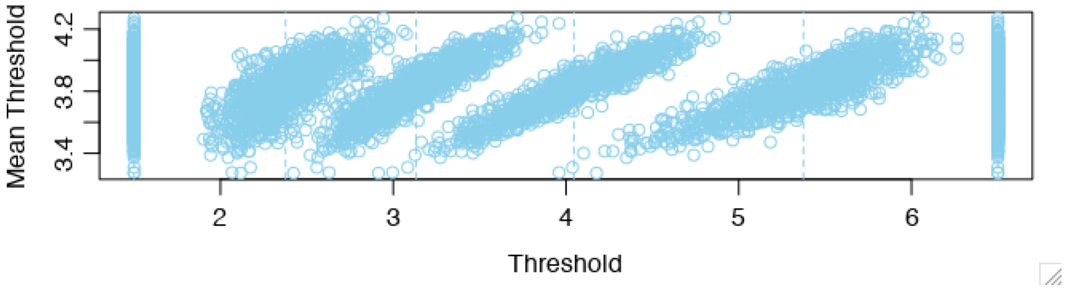



Intercept
mode = 53.9
95% HDI
26.8
89.2
20
40
60
80
100
120
β_0
Year
mode = -0.0274
95% HDI
-0.046
-0.0134
-0.06
-0.04
-0.02
β_1
Length
mode = 0.0405
95% HDI
0.0247
0.0588
0.01
0.03
0.05
0.07
β_2


Std.Dev.
mode = 1.11
95% HDI
0.465
1.37
0.4
0.8
1.2
1.6
σ



ii.) Is there anything unusual about the posterior distribution on the thresholds (see Figure 23.12), and why? Thresholds no longer get inverted; simply using a slightly different distribution to the normal and nothing more fundamental about the model changes as we had before.


Mean Threshold
Threshold