Introduction to regression models

Dr. Olanrewaju Michael Akande

March 27, 2020

Announcements

- Expect midterm key sometime today.

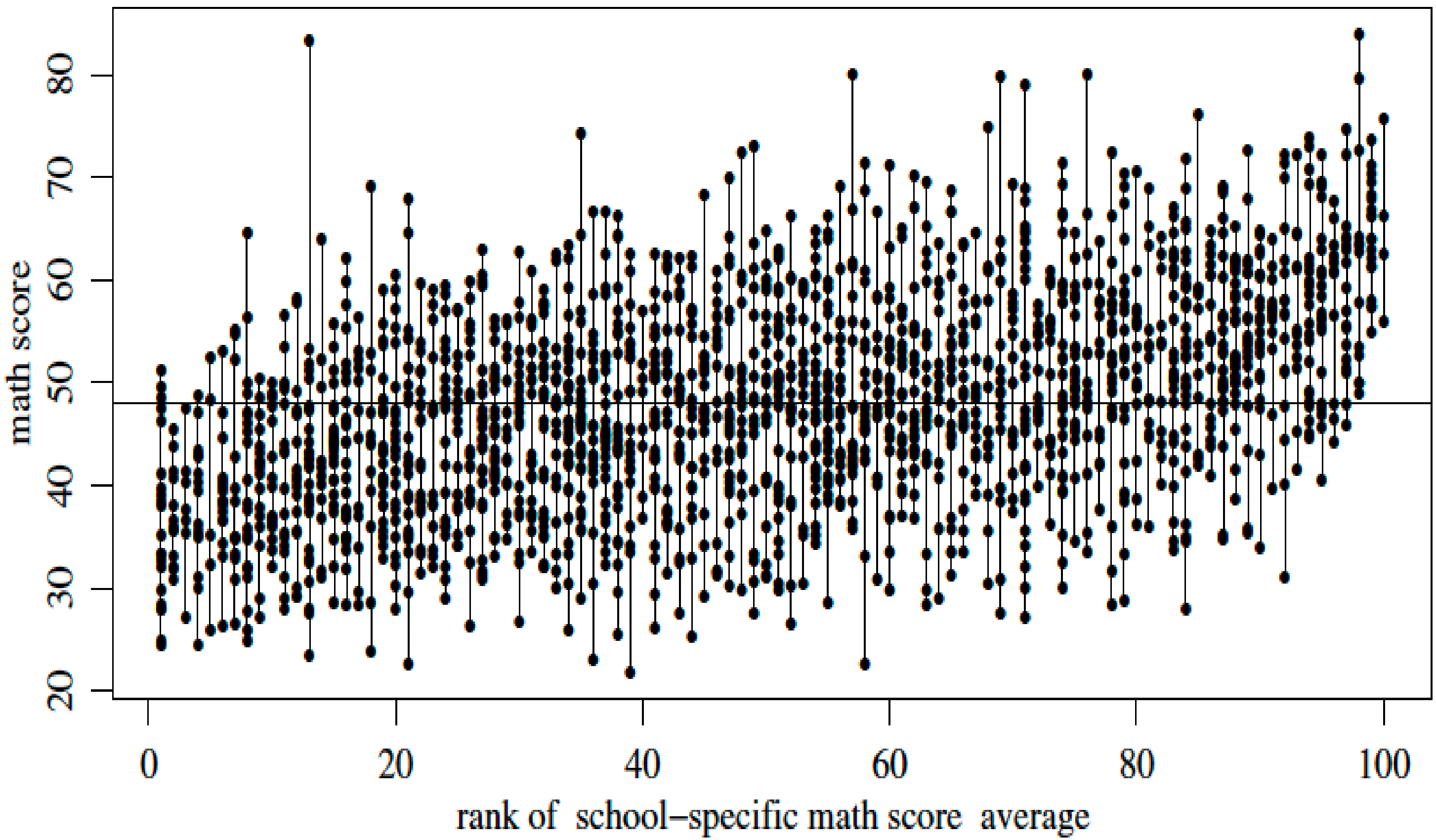
Outline

- Wrap up for hierarchical models
- Linear regression:
  - Motivating example
  - Frequentist estimation
  - Bayesian specification
  - Back to example

Wrap up for hierarchical models

ELS DATA

Recall the ELS data:



ELS HYPOTHESES

- Investigators may be interested in the following:
  - Differences in mean scores across schools
  - Differences in school-specific variances
- How do we evaluate these questions in a statistical model?

Hierarchical model

- Model:

$$
\begin{aligned}
y_{ij}|\theta_j, \sigma^2 &\sim \mathcal{N}\left(\theta_j, \sigma_j^2\right); \quad i = 1, \ldots, n_j \\
\theta_j|\mu, \tau^2 &\sim \mathcal{N}\left(\mu, \tau^2\right); \quad j = 1, \ldots, J \\
\sigma_1^2, \ldots, \sigma_J^2|\nu_0, \sigma_0^2 &\sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right) \\
\mu &\sim \mathcal{N}\left(\mu_0, \gamma_0^2\right) \\
\tau^2 &\sim \mathcal{IG}\left(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2}\right). \\
\pi(\nu_0) &\propto e^{-\alpha\nu_0} \\
\sigma_0^2 &\sim \mathcal{G}a\left(a, b\right).
\end{aligned}
$$

- Now, we need to specify hyperparameters. That should be fun!

Prior specification

- This exam was designed to have a national mean of 50 and standard deviation of 10. Suppose we don't have any other information.

- Then, we can specify

$$\mu \sim \mathcal{N}\left(\mu_0 = 50, \gamma_0^2 = 25\right)$$

$$\tau^2 \sim \mathcal{IG}\left(\frac{\eta_0}{2} = \frac{1}{2}, \frac{\eta_0 \tau_0^2}{2} = \frac{100}{2}\right).$$

$$\pi(\nu_0) \propto e^{-\alpha \nu_0} \propto e^{-\nu_0}$$

$$\sigma_0^2 \sim \mathcal{G}a\left(a = 1, b = \frac{1}{100}\right).$$

- Are these prior distributions overly informative?

Full conditionals (recap)

- $$\pi(\theta_j|\cdots\cdots) = \mathcal{N}\left(\mu_j^\star, \tau_j^\star\right) \quad \text{where}$$
$$\tau_j^\star = \frac{1}{\frac{n_j}{\sigma_j^2} + \frac{1}{\tau^2}}; \qquad \mu_j^\star = \tau_j^\star \left[\frac{n_j}{\sigma_j^2}\bar{y}_j + \frac{1}{\tau^2}\mu\right]$$

- $$\pi(\sigma_j^2|\cdots\cdots) = \mathcal{IG}\left(\frac{\nu_j^\star}{2}, \frac{\nu_j^\star \sigma_j^{2(\star)}}{2}\right) \quad \text{where}$$
$$\nu_j^\star = \nu_0 + n_j; \qquad \sigma_j^{2(\star)} = \frac{1}{\nu_j^\star}\left[\nu_0\sigma_0^2 + \sum_{i=1}^{n_j}(y_{ij} - \theta_j)^2\right].$$

- $$\pi(\mu|\cdots\cdots) = \mathcal{N}\left(\mu_n, \gamma_n^2\right) \quad \text{where}$$
$$\gamma_n^2 = \frac{1}{\frac{J}{\tau^2} + \frac{1}{\gamma_0^2}}; \qquad \mu_n = \gamma_n^2\left[\frac{J}{\tau^2}\bar{\theta} + \frac{1}{\gamma_0^2}\mu_0\right]$$

Full conditionals (recap)

- $$\pi(\tau^2|\cdots\cdots) = \mathcal{IG}\left(\frac{\eta_n}{2}, \frac{\eta_n \tau_n^2}{2}\right) \quad \text{where}$$

  $$\eta_n = \eta_0 + J; \qquad \tau_n^2 = \frac{1}{\eta_n}\left[\eta_0 \tau_0^2 + \sum_{j=1}^{J} (\theta_j - \mu)^2\right].$$

- $$\begin{aligned} \ln \pi(\nu_0|\cdots\cdots) \propto & \left(\frac{J\nu_0}{2}\right) \ln\left(\frac{\nu_0 \sigma_0^2}{2}\right) - J \ln\left[\Gamma\left(\frac{\nu_0}{2}\right)\right] \\ & + \left(\frac{\nu_0}{2} + 1\right)\left(\sum_{j=1}^{J} \ln\left[\frac{1}{\sigma_j^2}\right]\right) \\ & - \nu_0 \left[\alpha + \frac{\sigma_0^2}{2} \sum_{j=1}^{J} \frac{1}{\sigma_j^2}\right] \end{aligned}$$

- $$\pi(\sigma_0^2|\cdots\cdots) = \mathcal{G}a\left(\sigma_0^2; a_n, b_n\right) \quad \text{where}$$

  $$a_n = a + \frac{J\nu_0}{2}; \quad b_n = b + \frac{\nu_0}{2} \sum_{j=1}^{J} \frac{1}{\sigma_j^2}.$$

Side notes

- Obviously, as you have seen in the lab, we can simply use Stan (or JAGS, BUGS) to fit these models without needing to do any of this ourselves.
- The point here (as you should already know by now) is to learn and understand all the details, including the math!

Gibbs sampler

```r
#Data summaries
J <- length(unique(Y[,"school"]))
ybar <- c(by(Y[,"mathscore"],Y[,"school"],mean))
s_j_sq <- c(by(Y[,"mathscore"],Y[,"school"],var))
n <- c(table(Y[,"school"]))

#Hyperparameters for the priors
mu_0 <- 50
gamma_0_sq <- 25
eta_0 <- 1
tau_0_sq <- 100
alpha <- 1
a <- 1
b <- 1/100

#Grid values for sampling nu_0_grid
nu_0_grid<-1:5000

#Initial values for Gibbs sampler
theta <- ybar
sigma_sq <- s_j_sq
mu <- mean(theta)
tau_sq <- var(theta)
nu_0 <- 1
sigma_0_sq <- 100
```

Gibbs sampler

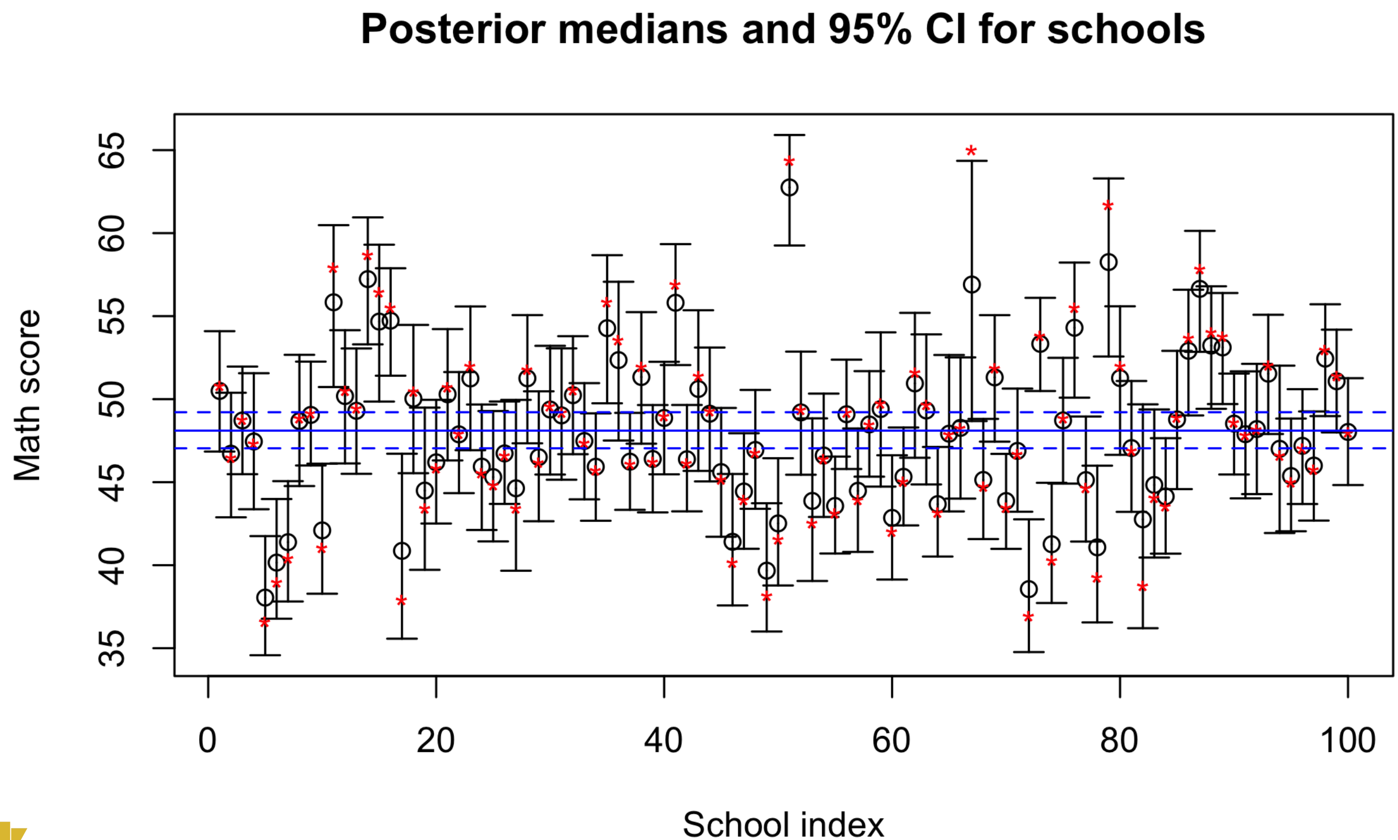
```
#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)


#Set null matrices to save samples
SIGMA_SQ <- THETA <- matrix(nrow=n_iter, ncol=J)
OTHER_PAR <- matrix(nrow=n_iter, ncol=4)


#Now, to the Gibbs sampler
for(s in 1:(n_iter+burn_in)){

  #update the theta vector (all the theta_j's)
  tau_j_star <- 1/(n/sigma_sq + 1/tau_sq)
  mu_j_star <- tau_j_star*(ybar*n/sigma_sq + mu/tau_sq)
  theta <- rnorm(J,mu_j_star,sqrt(tau_j_star))

  #update the sigma_sq vector (all the sigma_sq_j's)
  nu_j_star <- nu_0 + n
  theta_long <- rep(theta,n)
  nu_j_star_sigma_j_sq_star <-
    nu_0*sigma_0_sq + c(by((Y[,"mathscore"] - theta_long)^2,Y[,"school"],sum))
  sigma_sq <- 1/rgamma(J,(nu_j_star/2),(nu_j_star_sigma_j_sq_star/2))
  #update mu
  gamma_n_sq <- 1/(J/tau_sq + 1/gamma_0_sq)
  mu_n <- gamma_n_sq*(J*mean(theta)/tau_sq + mu_0/gamma_0_sq)
  mu <- rnorm(1,mu_n,sqrt(gamma_n_sq))
```

Gibbs sampler

```
  #update tau_sq
  eta_n <- eta_0 + J
  eta_n_tau_n_sq <- eta_0*tau_0_sq + sum((theta-mu)^2)
  tau_sq <- 1/rgamma(1,eta_n/2,eta_n_tau_n_sq/2)

  #update sigma_0_sq
  sigma_0_sq <- rgamma(1,(a + J*nu_0/2),(b + nu_0*sum(1/sigma_sq)/2))

  #update nu_0
  log_prob_nu_0 <- (J*nu_0_grid/2)*log(nu_0_grid*sigma_0_sq/2) -
    J*lgamma(nu_0_grid/2) +
    (nu_0_grid/2+1)*sum(log(1/sigma_sq)) -
    nu_0_grid*(alpha + sigma_0_sq*sum(1/sigma_sq)/2)
  nu_0 <- sample(nu_0_grid,1, prob = exp(log_prob_nu_0 - max(log_prob_nu_0)) )
  #this last step substracts the maximum logarithm from all logs
  #it is a neat trick that throws away all results that are so negative
  #they will screw up the exponential
  #note that the sample function will renormalize the probabilities internally


  #save results only past burn-in
  if(s > burn_in){
    THETA[(s-burn_in),] <- theta
    SIGMA_SQ[(s-burn_in),] <- sigma_sq
    OTHER_PAR[(s-burn_in),] <- c(mu,tau_sq,sigma_0_sq,nu_0)
  }
}
colnames(OTHER_PAR) <- c("mu","tau_sq","sigma_0_sq","nu_0")
```

Posterior inference for group means

The blue lines indicate the posterior median and a 95% for μ. The red asterisks indicate the data values $\bar{y}_j$.



Posterior inference for group variances

Posterior summaries of σ_j^2.

Posterior inference

Shrinkage as a function of sample size.

```
##    n Sample group mean Post. est. of group mean Post. est. of overall mean
## 1 31          50.81355                 50.49363                   48.10549
## 2 22          46.47955                 46.71544                   48.10549
## 3 23          48.77696                 48.71578                   48.10549
## 4 19          47.31632                 47.44935                   48.10549
## 5 21          36.58286                 38.04669                   48.10549
```

```
##     n Sample group mean Post. est. of group mean Post. est. of overall mean
## 15 12          56.43083                 54.67213                   48.10549
## 16 23          55.49609                 54.72904                   48.10549
## 17  7          37.92714                 40.86290                   48.10549
## 18 14          50.45357                 50.03007                   48.10549
```

```
##     n Sample group mean Post. est. of group mean Post. est. of overall mean
## 67  4          65.01750                 56.90436                   48.10549
## 68 19          44.74684                 45.13522                   48.10549
## 69 24          51.86917                 51.31079                   48.10549
## 70 27          43.47037                 43.86470                   48.10549
## 71 22          46.70455                 46.88374                   48.10549
## 72 13          36.95000                 38.55704                   48.10549
```

How about non-normal models?

- Suppose we have $y_{ij} \in \{0, 1, \ldots\}$ being a count for subject i in group j.
- For count data, it is natural to use a Poisson likelihood, that is,

  $$y_{ij} \sim \text{Poisson}(\theta_j)$$

  where each $\theta_j = \mathbb{E}[y_{ij}]$ is a group specific mean.
- When there are limited data within each group, it is natural to borrow information.
- How can we accomplish this with a hierarchical model?
- See homework 6 for a similar setup!

Linear regression model

Motivating example

- Let's consider the problem of predicting swimming times for high school swimmers to swim 50 yards.
- We have data collected on four students, each with six times taken (every two weeks).
- Suppose the coach of the team wants to use the data to recommend one of the swimmers to compete in a swim meet in two weeks time. Regression models sure seem like a good fit here.
- In a typical regression setup, we store the predictor variables in a matrix $\boldsymbol{X}_{n \times p}$, so n is the number of observations and p is the number of variables.
- You should all know how to write down and fit linear regression models of the most common forms, so let's only review the most important details.

Normal regression model

- The model assumes the following distribution for a response variable Y_i given multiple covariates/predictors $\boldsymbol{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{i(p-1)})$.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_{p-1} x_{i(p-1)} + \epsilon_i; \quad \epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

  or in vector form for the parameters,

$$Y_i = \boldsymbol{\beta}^T \boldsymbol{x}_i + \epsilon_i; \quad \epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2),$$

  where $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_{p-1})$.

- We can also write the model as:

$$Y_i \overset{iid}{\sim} \mathcal{N}(\boldsymbol{\beta}^T \boldsymbol{x}_i, \sigma^2);$$

$$p(y_i | \boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{\beta}^T \boldsymbol{x}_i, \sigma^2).$$

- That is, the model assumes $\mathbb{E}[Y|\boldsymbol{x}]$ is linear.

Likelihood

- Given that we have $Y_i \overset{iid}{\sim} \mathcal{N}(\boldsymbol{\beta}^T \boldsymbol{x}_i, \sigma^2)$, the likelihood is

$$
\begin{aligned}
p(y_i, \ldots, y_n | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p, \boldsymbol{\beta}, \sigma^2) &= \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 \right\} \\
&\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 \right\}.
\end{aligned}
$$

- From all our work with normal models, we already know it would be convenient to specify a (multivariate) normal prior on $\boldsymbol{\beta}$ and a gamma prior on $1/\sigma^2$, so let's start there.

- Two things to immediately notice:

  - since $\boldsymbol{\beta}$ is a vector, it might actually be better to rewrite this kernel in multivariate form altogether, and

  - when combining this likelihood with the prior kernel, we will need to find a way to detach $\boldsymbol{\beta}$ from $\boldsymbol{x}_i$.

Multivariate form

- Let

$$\boldsymbol{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \quad \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1(p-1)} \\ 1 & x_{21} & x_{22} & \dots & x_{2(p-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{n(p-1)} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad \boldsymbol{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- Then, we can write the model as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \;\; \boldsymbol{\epsilon} \sim \mathcal{N}_n(0, \sigma^2 \boldsymbol{I}_{n\times n}).$$

- That is, in multivariate form, we have

$$\boldsymbol{Y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_{n\times n}).$$

Frequentist estimation recap

- OLS estimate of β is given by

$$\hat{\boldsymbol{\beta}}_{\text{ols}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

- Predictions can then be written as

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\text{ols}} = \boldsymbol{X}\left[\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}\right] = \left[\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\right]\boldsymbol{y}.$$

- The variance of the OLS estimates of all p coefficients is

$$\mathbb{V}ar\left[\hat{\boldsymbol{\beta}}_{\text{ols}}\right] = \sigma^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}.$$

- Finally,

$$s_e^2 = \frac{(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\text{ols}})^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\text{ols}})}{n-p}.$$

Bayesian specification

Bayesian specification

- Now, our likelihood becomes

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right\} \\ &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\right]\right\}. \end{aligned}$$

- We can start with the following semi-conjugate prior for β:

$$\pi(\boldsymbol{\beta}) = \mathcal{N}_p(\boldsymbol{\beta}_0, \Sigma_0).$$

- That is, the pdf is

$$\pi(\boldsymbol{\beta}) = (2\pi)^{-\frac{p}{2}}|\Sigma_0|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^T\Sigma_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right\}.$$

- Recall from our multivariate normal model that we can write this pdf as

$$\pi(\boldsymbol{\beta}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\beta} + \boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\mu}_0\right\}.$$

Multivariate normal model recap

- To avoid doing all work from scratch, we can leverage results from the multivariate normal model.

- In particular, recall that if $\boldsymbol{Y} \sim \mathcal{N}_p(\boldsymbol{\theta}, \Sigma)$,

$$p(\boldsymbol{y}|\boldsymbol{\theta}, \Sigma) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\theta}^T(\Sigma^{-1})\boldsymbol{\theta} + \boldsymbol{\theta}^T(\Sigma^{-1}\bar{\boldsymbol{y}})\right\}$$

and

$$\pi(\boldsymbol{\theta}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\theta}^T\Lambda_0^{-1}\boldsymbol{\theta} + \boldsymbol{\theta}^T\Lambda_0^{-1}\boldsymbol{\mu}_0\right\}$$

- Then

$$\pi(\boldsymbol{\theta}|\Sigma, \boldsymbol{y}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\theta}^T\left[\Lambda_0^{-1} + \Sigma^{-1}\right]\boldsymbol{\theta} + \boldsymbol{\theta}^T\left[\Lambda_0^{-1}\boldsymbol{\mu}_0 + \Sigma^{-1}\bar{\boldsymbol{y}}\right]\right\} \equiv \mathcal{N}_p(\boldsymbol{\mu}_n, \Lambda_n)$$

where

$$\begin{aligned}\Lambda_n &= \left[\Lambda_0^{-1} + \Sigma^{-1}\right]^{-1}\\ \boldsymbol{\mu}_n &= \Lambda_n\left[\Lambda_0^{-1}\boldsymbol{\mu}_0 + \Sigma^{-1}\bar{\boldsymbol{y}}\right].\end{aligned}$$

Posterior computation

- For inference on β, rewrite the likelihood as

$$
\begin{aligned}
p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\beta},\sigma^2) &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\left[\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\left(\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{\beta} + \boldsymbol{\beta}^T\left(\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y}\right)\right\}.
\end{aligned}
$$

- Again, with the prior written as

$$
\pi(\boldsymbol{\beta}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\beta} + \boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\mu}_0\right\},
$$

both forms look like what we have on the previous page. It is then easy to read off the full conditional for β.

Posterior computation

- That is,

$$\begin{aligned}\pi(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\sigma^2) &\propto p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\beta},\sigma^2)\cdot\pi(\boldsymbol{\beta})\\ &\propto \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\left[\Sigma_0^{-1}+\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}\right]\boldsymbol{\beta}+\boldsymbol{\beta}^T\left[\Sigma_0^{-1}\boldsymbol{\beta}_0+\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y}\right]\right\}\\ &\equiv\ \mathcal{N}_p(\boldsymbol{\mu}_n,\Sigma_n).\end{aligned}$$

- Comparing this to the prior

$$\pi(\boldsymbol{\beta})\propto\exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\beta}+\boldsymbol{\beta}^T\Sigma_0^{-1}\boldsymbol{\mu}_0\right\},$$

means

$$\begin{aligned}\Sigma_n &= \left[\Sigma_0^{-1}+\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}\right]^{-1}\\ \boldsymbol{\mu}_n &= \Sigma_n\left[\Sigma_0^{-1}\boldsymbol{\beta}_0+\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y}\right].\end{aligned}$$

Posterior computation

- Next, we move to σ^2. From previous work, we already know the inverse-gamma distribution with be semi-conjugate.

- First, recall that $\mathcal{IG}(y; a, b) \equiv \dfrac{b^a}{\Gamma(a)} y^{-(a+1)} e^{-\dfrac{b}{y}}$.

- So, if we set $\pi(\sigma^2) = \mathcal{IG}\left(\dfrac{\nu_0}{2}, \dfrac{\nu_0 \sigma_0^2}{2}\right)$, we have

$$
\begin{aligned}
\pi(\sigma^2 | \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\beta}) &\propto p(\boldsymbol{y} | \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) \cdot \pi(\sigma^2) \\
&\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{ - \left(\frac{1}{\sigma^2}\right) \frac{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})}{2} \right\} \\
&\quad \times (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} e^{-\left(\frac{1}{\sigma^2}\right)\left[\frac{\nu_0 \sigma_0^2}{2}\right]}
\end{aligned}
$$

Posterior computation

- That is,

$$\begin{aligned}\pi(\sigma^2|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\beta}) &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{ -\left(\frac{1}{\sigma^2}\right) \frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2} \right\} \\ &\quad \times (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} e^{-\left(\frac{1}{\sigma^2}\right)\left[\frac{\nu_0\sigma_0^2}{2}\right]} \\ &\propto (\sigma^2)^{-\left(\frac{\nu_0+n}{2}+1\right)} e^{-\left(\frac{1}{\sigma^2}\right)\left[\frac{\nu_0\sigma_0^2 + (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2}\right]} \\ &\equiv \mathcal{IG}\left(\frac{\nu_n}{2}, \frac{\nu_n\sigma_n^2}{2}\right),\end{aligned}$$

where

$$\nu_n = \nu_0 + n; \quad \sigma_n^2 = \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right] = \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + \mathrm{SSR}(\boldsymbol{\beta})\right].$$

- $(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$ is the sum of squares of the residuals (SSR).

Swimming data

- Back to the swimming example. The data is from Exercise 9.1 in Hoff.
- The data set we consider contains times (in seconds) of four high school swimmers swimming 50 yards.

```
Y <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/swim.dat")
Y
```

```
##     V1   V2   V3   V4   V5   V6
## 1 23.1 23.2 22.9 22.9 22.8 22.7
## 2 23.2 23.1 23.4 23.5 23.5 23.4
## 3 22.7 22.6 22.8 22.8 22.9 22.8
## 4 23.7 23.6 23.7 23.5 23.5 23.4
```

- There are 6 times for each student, taken every two weeks. That is, each swimmer has six measurements at $t = 2, 4, 6, 8, 10, 12$ weeks.
- Each row corresponds to a swimmer and a higher column index indicates a later date.

Swimming data

- Given that we don't have enough data, we can explore hierarchical models (just as in the lab). That way, we can borrow information across swimmers.

- For now, however, we will fit a separate linear regression model for each swimmer, with swimming time as the response and week as the explanatory variable (which we will mean center).

- For setting priors, we have one piece of information: times for this age group tend to be between 22 and 24 seconds.

- Based on that, we can set uninformative parameters for the prior on σ^2 and for the prior on $\boldsymbol{\beta}$, we can set

$$\pi(\boldsymbol{\beta}) = \mathcal{N}_2 \left(\boldsymbol{\beta}_0 = \begin{pmatrix} 23 \\ 0 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix} \right).$$

- This centers the intercept at 23 (the middle of the given range) and the slope at 0 (so we are assuming no increase) but we choose the variance to be a bit large to err on the side of being less informative.

Posterior computation

```r
#Create X matrix, transpose Y for easy computayion
Y <- t(Y)
n_swimmers <- ncol(Y)
n <- nrow(Y)
W <- seq(2,12,length.out=n)
X <- cbind(rep(1,n),(W-mean(W)))
p <- ncol(X)

#Hyperparameters for the priors
beta_0 <- matrix(c(23,0),ncol=1)
Sigma_0 <- matrix(c(5,0,0,2),nrow=2,ncol=2)
nu_0 <- 1
sigma_0_sq <- 1/10

#Initial values for Gibbs sampler
#No need to set initial value for sigma^2, we can simply sample it first
beta <- matrix(c(23,0),nrow=p,ncol=n_swimmers)
sigma_sq <- rep(1,n_swimmers)

#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

#Set null matrices to save samples
BETA <- array(0,c(n_swimmers,n_iter,p))
SIGMA_SQ <- matrix(0,n_swimmers,n_iter)
```

Posterior computation

```r
#Now, to the Gibbs sampler
#library(mvtnorm) for multivariate normal

#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

for(s in 1:(n_iter+burn_in)){
  for(j in 1:n_swimmers){

    #update the sigma_sq
    nu_n <- nu_0 + n
    SSR <- t(Y[,j] - X%*%beta[,j])%*%(Y[,j] - X%*%beta[,j])
    nu_n_sigma_n_sq <- nu_0*sigma_0_sq + SSR
    sigma_sq[j] <- 1/rgamma(1,(nu_n/2),(nu_n_sigma_n_sq/2))

    #update beta
    Sigma_n <- solve(solve(Sigma_0) + (t(X)%*%X)/sigma_sq[j])
    mu_n <- Sigma_n %*% (solve(Sigma_0)%*%beta_0 + (t(X)%*%Y[,j])/sigma_sq[j])
    beta[,j] <- rmvnorm(1,mu_n,Sigma_n)

    #save results only past burn-in
    if(s > burn_in){
      BETA[j,(s-burn_in),] <- beta[,j]
      SIGMA_SQ[j,(s-burn_in)] <- sigma_sq[j]
    }
  }
}
```

Results

- Before looking at the posterior samples, what are the OLS estimates for all the parameters?

```r
beta_ols <- matrix(0,nrow=p,ncol=n_swimmers)
for(j in 1:n_swimmers){
beta_ols[,j] <- solve(t(X)%*%X)%*%t(X)%*%Y[,j]
}
colnames(beta_ols) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(beta_ols) <- c("beta_0","beta_1")
beta_ols
```

```
##           Swimmer 1   Swimmer 2 Swimmer 3   Swimmer 4
## beta_0 22.93333333 23.35000000  22.76667 23.56666667
## beta_1 -0.04571429  0.03285714   0.02000 -0.02857143
```

- Give an interpretation for the parameters.
- Any thoughts on who the coach should recommend based on this alone?
- Is this how we should be answering the question?

Posterior inference

- Posterior means are almost identical to OLS estimates.

```
beta_postmean <- t(apply(BETA,c(1,3),mean))
colnames(beta_postmean) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(beta_postmean) <- c("beta_0","beta_1")
beta_postmean
```

```
##         Swimmer 1   Swimmer 2   Swimmer 3   Swimmer 4
## beta_0 22.9339174 23.34963191 22.76617785 23.56614309
## beta_1 -0.0453998  0.03251415  0.01991469 -0.02854268
```

- How about confidence intervals?

```
beta_postCI <- apply(BETA,c(1,3),function(x) quantile(x,probs=c(0.025,0.975)))
colnames(beta_postCI) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_postCI[,,1]; beta_postCI[,,2]
```

```
##       Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## 2.5%   22.76901  23.15949  22.60097  23.40619
## 97.5%  23.09937  23.53718  22.93082  23.73382
```

```
##          Swimmer 1   Swimmer 2   Swimmer 3   Swimmer 4
## 2.5%  -0.093131856 -0.02128792 -0.02960257 -0.07704344
## 97.5%  0.002288246  0.08956464  0.06789081  0.01940960
```

- Is there any evidence that the times matter?

Posterior inference

- Is there any evidence that the times matter?

```
beta_pr_great_0 <- t(apply(BETA,c(1,3),function(x) mean(x > 0)))
colnames(beta_pr_great_0) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_pr_great_0
```

```
##      Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## [1,]    1.0000    1.0000    1.0000    1.0000
## [2,]    0.0287    0.9044    0.8335    0.0957
```

```
#or alternatively,
beta_pr_less_0 <- t(apply(BETA,c(1,3),function(x) mean(x < 0)))
colnames(beta_pr_less_0) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_pr_less_0
```

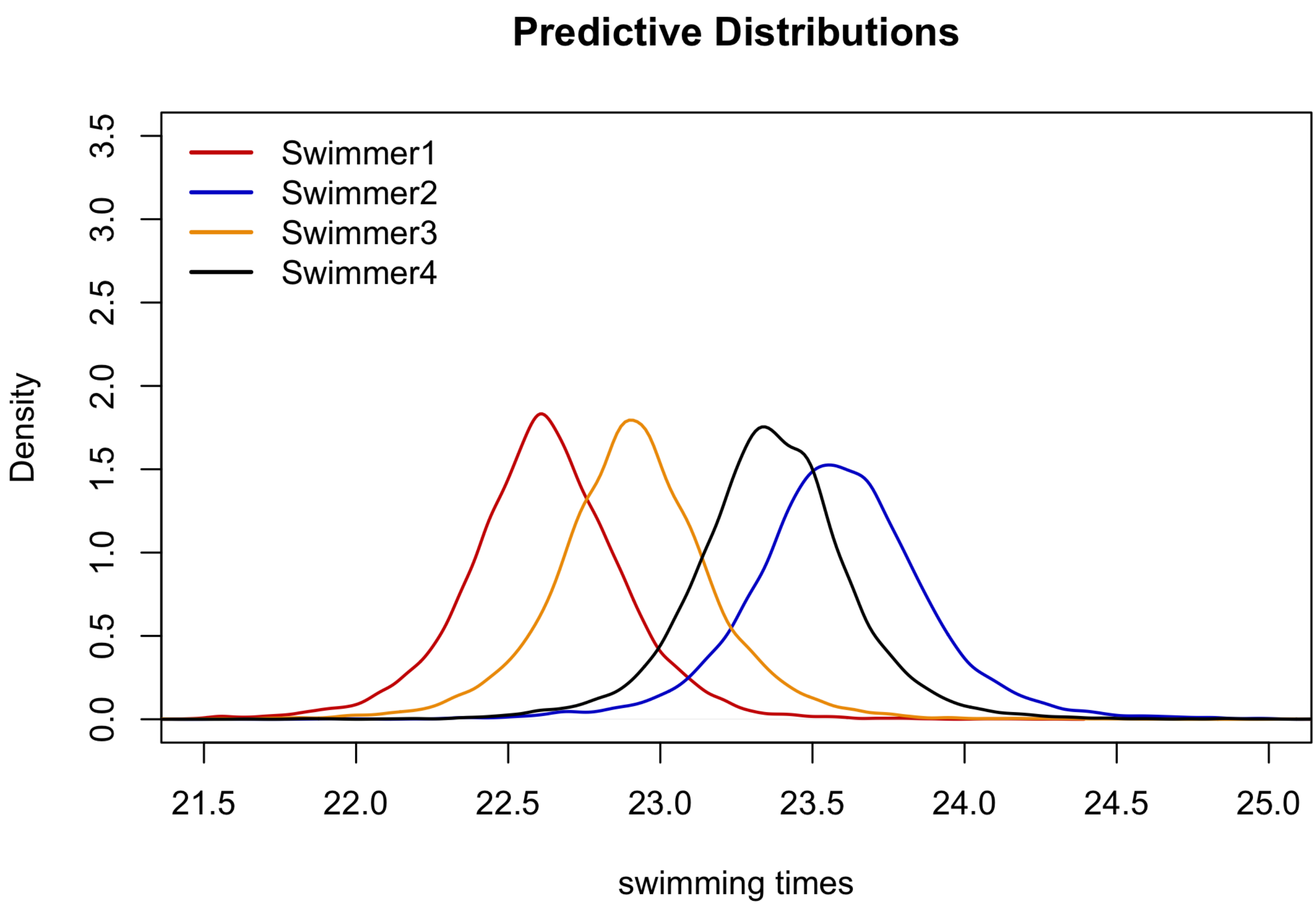
```
##      Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## [1,]    0.0000    0.0000    0.0000    0.0000
## [2,]    0.9713    0.0956    0.1665    0.9043
```

Posterior predictive inference

- How about the posterior predictive distributions for a future time two weeks after the last recorded observation?

```
x_new <- matrix(c(1,(14-mean(W))),ncol=1)
post_pred <- matrix(0,nrow=n_iter,ncol=n_swimmers)
for(j in 1:n_swimmers){
post_pred[,j] <- rnorm(n_iter,BETA[j,,]%*%x_new,sqrt(SIGMA_SQ[j,]))
}
colnames(post_pred) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")

plot(density(post_pred[,"Swimmer 1"]),col="red3",xlim=c(21.5,25),ylim=c(0,3.5),lwd=1.5
 main="Predictive Distributions",xlab="swimming times")
legend("topleft",2,c("Swimmer1","Swimmer2","Swimmer3","Swimmer4"),col=c("red3","blue3"
lines(density(post_pred[,"Swimmer 2"]),col="blue3",lwd=1.5)
lines(density(post_pred[,"Swimmer 3"]),col="orange2",lwd=1.5)
lines(density(post_pred[,"Swimmer 4"]),lwd=1.5)
```

Posterior predictive inference



Posterior predictive inference

- How else can we answer the question on who the coach should recommend for the swim meet in two weeks time? Few different ways.

- Let $Y_j^\star$ be the predicted swimming time for each swimmer j. We can do the following: using draws from the predictive distributions, compute the posterior probability that $P(Y_j^\star = \min(Y_1^\star, Y_2^\star, Y_3^\star, Y_4^\star))$ for each swimmer j, and based on this make a recommendation to the coach.

- That is,

```
post_pred_min <- as.data.frame(apply(post_pred,1,function(x) which(x==min(x))))
colnames(post_pred_min) <- "Swimmers"
post_pred_min$Swimmers <- as.factor(post_pred_min$Swimmers)
levels(post_pred_min$Swimmers) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
table(post_pred_min$Swimmers)/n_iter
```

```
##
## Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
##    0.7790    0.0078    0.1994    0.0138
```

- Which swimmer would you recommend?