SDS 383D Ex 01:
Preliminaries

January 18, 2016

Jennifer Starling

Bayesian Inference in Simple Conjugate Families

Part A

Let $x_1, ..., x_n \sim$ iid Bernoulli(w). Let $w \sim$ Beta(a,b) be the prior.

Let y be the number of successes in the sequence of n Bernoulli trials. Then $y \sim Binom(n, w)$.

We begin with the following pdfs:

$$\text{Prior is } p(w) = \frac{1}{Beta(a,b)} w^{a-1}(1-w)^{b-1}$$
$$\text{Sampling model is } p(y|w) = \binom{n}{y} w^y (1-w)^{n-y}$$

Then $posterior \propto sampling * prior$.

$$\begin{aligned} p(w|y) &\propto w^y(1-w)^{n-y} * w^{a-1}(1-w)^{b-1} \\ &= w^{a+y-1}(1-w)^{b+(n-y)-1} \end{aligned}$$

This is the kernel of the $Beta(a+y, b+n-y)$ distribution.

The posterior is $p(w|y) \sim Beta(a+y, b+n-y)$. ■

Part B

The pdf for the $gamma(a, b)$ distribution is: $p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$.

Let $x_1 \sim gamma(a_1, 1)$ and $x_2 \sim gamma(a_2, 1)$. Define $y_1 = \frac{x_1}{x_1+x_2}$ and $y_2 = x_1 + x_2$.

First, obtain the joint density of (y_1, y_2) using the standard bivariate transformation procedure (as defined in Chapter 4 of Casella and Berger).

Step 1: Obtain Transformation Equations

Find $g_1^{-1}(x_1, x_2)$ and $g_2^{-1}(x_1, x_2)$ inverse equations, and check that transformation is 1-1 and onto.

$$
\begin{aligned}
&y_1 = \frac{x_1}{x_1 + x_2} \text{ and } y_2 = x_1 + x_2 \\
&y_1 = \frac{x_1}{x_2} \text{ (plug 2nd equation into 1st)} \\
&x_1 = y_1 y_2 \rightarrow g_1^{-1}(y_1, y_2) = y_1 y_2 \\
&\text{Plug previous result for } x_1 \text{ into second equation.} \\
&y_2 = y_1 y_2 + x_2 \rightarrow x_2 = y_2 - y_1 y_2 \rightarrow g_2^{-1}(y_1, y_2) = y_2 - y_1 y_2
\end{aligned}
$$

This transformation is 1-1 and onto, with support mapping $\{x_1 > 0, x_2 > 0\} \rightarrow \{0 < y_1 < 1, y_2 > 0\}$.

- Onto: Met since able to find unique inverse equations in part 1, above.
- 1-1. Met. Let $(y_{11}, y_{21}) = (y_{21}, y_{22})$. We can then do the algebra to show that $(x_{11}, x_{21}) = (x_{21}, y_{22})$.

Step 2: Jacobian

$$
\begin{aligned}
|J| &= \left| \begin{pmatrix} \frac{\delta g_1^{-1}}{\delta y_1} & \frac{\delta g_1^{-1}}{\delta y_2} \\ \frac{\delta g_2^{-1}}{\delta y_1} & \frac{\delta g_1^{-1}}{\delta y_2} \end{pmatrix} \right| \\
&= \left| \begin{pmatrix} y_2 & y_1 \\ -y_2 & (1 - y_1) \end{pmatrix} \right| \\
&= |y_2(1 - y_1)| + y_1 y_2 \\
&= |y_2 - y_1 y_2| \\
&= |y_2| \\
&= y_2 \text{ since } y_2 > 0
\end{aligned}
$$

Therefore, $|J| = y_2$.

Step 3: Joint pdf

Since $x_1 \perp x_2$, the joint pdf of x_1 and x_2 is:

$f_{x_1,x_2}(x_1,x_2) = f(x_1)f(x_2) = \frac{1}{\Gamma(a_1)\Gamma(a_2)} x_1^{a_1-1} x_2^{a_2-1} e^{(-x_1-x_2)}$.

The joint pdf of y_1 and y_2 is:

$$\begin{aligned} f_{x_1,x_2}(g_1^{-1}, g_2^{-1})|J| &= \frac{1}{\Gamma(a_1)\Gamma(a_2)} (y_1y_2)^{a_1-1}(y_2 - y_1y_2)^{a_2-1} e^{\{-y_1y_2 - y_2(1-y_1)\}} y_2 \\ &= \frac{1}{\Gamma(a_1)\Gamma(a_2)} (y_1y_2)^{a_1-1}(y_2 - y_1y_2)^{a_2-1} e^{(y_2)} y_2 \end{aligned}$$

The joint pdf can be factored into functions of y_1 and y_2 as follows. We can also multiply and divide by $\Gamma(a_1 + a_2)$ to make it easy to identify the marginal densities.

$$\begin{aligned} &= \frac{1}{\Gamma(a_1+a_2)} \frac{\Gamma(a_1+a_2)}{\Gamma(a_1)\Gamma(a_2)} y_1^{a_1-1} y_2^{a_1-1} y_2^{a_2-1} (1-y_1)^{a_2-1} y_2 e^{(-y_2)} \\ &= \left[\frac{\Gamma(a_1+a_2)}{\Gamma(a_1)\Gamma(a_2)} y_1^{a_1-1}(1-y_1)^{a_2-1}\right] \left[\frac{1}{\Gamma(a_1+a_2)} y_2^{a_1+a_2-1} e^{(-y_2)}\right] \end{aligned}$$

These are the forms of the beta and gamma densities, respectively.

- $y_1 \sim Beta(a_1, a_2)$
- $y_2 \sim Gamma(a_1 + a_2, 1)$

We can then devise a process to generate Beta realizations. We can generate two independent gamma realizations (x_1, x_2) and calculate $y_1 = \frac{x_1}{x_1+x_2}$ to simulate the Beta realizations.

Part C

Let $x_1, ..., x_N \sim N(\theta, \sigma^2)$ where θ is unknown and σ^2 is known. The prior for θ is $\theta \sim N(m, v)$. Derive the posterior for $p(\theta|x_1, ..., x_N)$.

Prior:

$$p(\theta) = \frac{1}{\sqrt{2\pi v}} exp\{-\frac{1}{2v}(\theta - m)^2\}$$

Sampling model:

$$\begin{aligned} p(x_1, ..., x_n|\theta, \sigma^2) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{1}{2\sigma^2}(x_i - \theta)^2\} \\ &= (2\pi\sigma^2)^{-n/2} exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \theta)^2\} \end{aligned}$$

Expand the summation in the exponential term to make it easier to work with:

$$\sum_{i=1}^{n}(x_i - \theta)^2 = \sum_{i=1}^{n}(x_i - \theta)(x_i - \theta) = \sum_{i=1}^{n}[x_i^2 - 2\theta\sum_{i=1}^{n}x_i + n\theta^2] = n\bar{x}^2 - 2n\theta\bar{x} + n\theta^2$$

Then $posterior \propto sampling * prior$:

$$p(\theta|x_1, ..., x_n) \propto exp\left\{-\frac{1}{2\sigma^2}\left(n\bar{x}^2 - 2n\theta\bar{x} + n\theta^2\right)\right\} * exp\left\{-\frac{1}{2v}\left(\theta^2 - 2\theta m + m^2\right)\right\}$$

Drop all terms unrelated to θ (remember, σ^2 is known, so is okay). Combine into one exponential term.

$$= exp\left\{-\frac{1}{2}\left(\frac{n}{\sigma^2}\theta^2 + \frac{1}{v}\theta^2 - \frac{2n\bar{x}}{\sigma^2}\theta - \frac{2m}{v}\theta\right)\right\}$$

Combine the θ^2 coefficients and the θ coefficients to make this form easier to work with. Let:

$$\begin{aligned} a &= \left(\frac{n}{\sigma^2} + \frac{1}{v}\right) \\ b &= \left(\frac{2n\bar{x}}{\sigma^2} + \frac{2m}{v}\right) \end{aligned}$$

This yields the equation $= exp\left\{-\frac{1}{2}\left(a\theta^2 - 2b\theta\right)\right\}$. Now we need to complete the square.

> Aside: A brief refresher on completing the square.
>
> - Begin with $ax^2 - 2bx$. Need form $x^2 - 2bx + b^2$, since this factors into $(x + b)^2$.
> - Accomplish this by factoring out a to obtain $a(x^2 - 2\frac{b}{a}x)$.
> - Then add and subtract $(\frac{b}{a})^2$ inside the parenthesis.

In our case, begin working with just the exponential term for completing the square.

$$exp\left\{-\frac{1}{2}\left(a\theta^2 - 2b\theta\right)\right\}$$

Factor out a.

$$-\frac{a}{2}\left(\theta^2 - 2\frac{b}{a}\theta\right)$$

Add and subtract $(\frac{b}{a})^2$ inside the parenthesis to get

$$-\frac{a}{2}\left(\theta^2 - 2\frac{b}{a}\theta + (\frac{b}{a})^2 - (\frac{b}{a})^2\right)$$

The added and subtracted terms are not functions of θ, so we can drop the $-(\frac{b}{a})^2$ term, leaving

$$\begin{aligned}&-\frac{a}{2}\left(\theta^2 - 2\frac{b}{a}\theta + (\frac{b}{a})^2\right)\\ &= -\frac{a}{2}(\theta - \frac{b}{a})^2\end{aligned}$$

Plug the exponential term back into the full equation

$$exp\left\{-\frac{a}{2}(\theta - \frac{b}{a})^2\right\}$$

This has the form of a normal distribution, with mean $\frac{b}{a}$ and variance $\frac{1}{a}$, ie precision equals a.

The posterior is

$$\begin{aligned}p(\theta|x_1, ..., x_n) &\sim N\left[\left(\frac{2n\bar{x}}{\sigma^2} + \frac{2m}{v}\right), \left(\frac{n}{\sigma^2} + \frac{1}{v}\right)^{-1}\right]\\ &= N\left(\frac{\left(\frac{m}{v} + \frac{\sum_{i=1}^{n}}{\sigma^2}\right)}{\left(\frac{1}{v} + \frac{n}{\sigma^2}\right)}, \left(\frac{1}{v} + \frac{n}{\sigma^2}\right)^{-1}\right)\end{aligned}$$

Note regarding intuition:

The second way is a more intuitive way to write the posterior parameters, since the mean is a precision-weighted average of the prior mean and the sample mean of the data. The precision is additive. It is often easier to work with precisions than variances.

Part D

Let $x_1, ..., x_n \sim N(\theta, \sigma^2)$ where θ is known and σ^2 is unknown. Will express σ^2 in terms of precison $w = \frac{1}{\sigma^2}$. Find the posterior $p(w|x_1, ..., x_n)$.

Prior for w is $w \sim Gamma(a, b)$. Identical to $\sigma^2 \sim IG(a, b)$.

$$p(w) = \frac{b^a}{\Gamma(a)} w^{a-1} e^{-bw}$$

Sampling model:

$$\begin{aligned} p(x_1, ..., x_n|\theta, w) &= \prod_{i=1} n \left(\frac{w}{2\pi}\right)^{\frac{1}{2}} exp\left\{-\frac{w}{2}(x_i - \theta)^2\right\} \\ &= \frac{w^{n/2}}{(2\pi)^{n/2}} exp\left\{-\frac{w}{2}\sum_{i=1}^{n}(x_i - \theta)^2\right\} \end{aligned}$$

Then $posterior \propto sampling * prior$:

$$p(w|x_1, ..., x_n) \propto w^{a+\frac{n}{2}-1} exp\left\{-w\left(b + \frac{\sum_{i=1}^{n}(x_i - \theta)^2}{2}\right)\right\}$$

This is the form of the gamma distribution, so the posterior for w is

$$p(w|x_1, ..., x_n) \sim Gamma\left(a + \frac{n}{2}, b + \frac{\sum_{i=1}^{n}(x_i - \theta)^2}{2}\right)$$

Equivalently, the posterior for σ^2 is Inverse Gamma (IG), with the same parameters.

Part E

Let $x_1, ..., x_n \sim N(\theta, \sigma_i^2)$ where θ is common for all x_i and is unknown. Variances are unique for each x_i and are known. The prior is $\theta \sim N(m, v)$. Derive the posterior for $p(\theta|x_1, ..., x_n)$.

Prior:

$$p(\theta) = \frac{1}{\sqrt{2\pi v}} exp\{-\frac{1}{2v}(\theta - m)^2\}$$

Sampling Model:

$$p(x_1, ..., x_n|\theta, \sigma_1^2, ..., \sigma_n^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\{-\frac{1}{2\sigma_i^2}(x_i - \theta)^2\}$$

Drop the constant of proportionality from the sampling model since it does not depend on θ.

$$\propto exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}(x_i - \theta)^2\right\}$$

Then $posterior \propto sampling * prior$:

$$
\begin{aligned}
p(\theta|x_1,...,x_n) &\propto exp\left\{-\frac{1}{2v}(\theta-m)^2 - \frac{1}{2}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}(x_i-\theta)^2\right\} \\
&= exp\left\{-\frac{1}{2}\left[\frac{(\theta-m)^2}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}(x_i-\theta)^2\right]\right\} \\
&= exp\left\{-\frac{1}{2}\left[\frac{\theta^2}{v} - \frac{2m}{v}\theta + \frac{m^2}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}(x_i-\theta)(x_i-\theta)\right]\right\} \\
&= exp\left\{-\frac{1}{2}\left[\frac{\theta^2}{v} - \frac{2m}{v}\theta + \frac{m^2}{v} + \sum_{i=1}^{n}\frac{x_i^2}{\sigma_i^2} - 2\theta\sum_{i=1}^{n}\frac{x_i}{\sigma_i^2} + \theta^2\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right]\right\} \\
&\propto exp\left\{-\frac{1}{2}\left[\frac{\theta^2}{v} - \frac{2m}{v}\theta - 2\theta\sum_{i=1}^{n}\frac{x_i}{\sigma_i^2} + \theta^2\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right]\right\}, \text{as} \frac{m^2}{v} \text{ and } \sum_{i=1}^{n}\frac{x_i^2}{\sigma_i^2} \text{ don't depend on } \theta \\
&= exp\left\{-\frac{1}{2}\left[\theta^2\left(\frac{1}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right) - 2\theta\left(\frac{m}{v} + \sum_{i=1}^{n}\frac{x_i}{\sigma_i^2}\right)\right]\right\}, \text{by grouping } \theta^2 \text{ and } \theta \text{ terms}
\end{aligned}
$$

Then, as before, we can use a and b to facilitate completing the square. Let

$$
\begin{aligned}
a &= \left(\frac{1}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right) \\
b &= \left(\frac{m}{v} + \sum_{i=1}^{n}\frac{x_i}{\sigma_i^2}\right)
\end{aligned}
$$

Then we have $exp\left\{-\frac{1}{2}[a\theta^2 - 2b\theta]\right\}$. We can repeat the process from Part C to complete the square. Working with just the inside term of the exponential expression:

$$a\theta^2 - 2b\theta = -\frac{a}{2}[\theta^2 - 2\frac{b}{a}\theta + \frac{b^2}{a^2} - \frac{b^2}{a^2}] = -\frac{a}{2}(\theta - \frac{b}{a})^2$$

Plugging back into the exponential, we have

$$exp\left\{-\frac{a}{2}(\theta - \frac{b}{a})^2\right\}$$

This is the form of the normal density with mean $\frac{a}{b}$ and precision a. Therefore, the posterior is distributed as follows:

$$p(\theta|x_1,...,x_n) \sim N\left(\frac{1}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}, \frac{\frac{m}{v} + \sum_{i=1}^{n}\frac{x_i}{\sigma_i^2}}{\frac{1}{v} + \sum_{i=1}^{n}\frac{1}{\sigma_i^2}}\right)$$

Part F

Let $(x|\sigma^2) \sim N(0, \sigma^2)$ with prior $\frac{1}{\sigma^2} \sim Gamma(a, b)$, as in part D. Show the marginal of x is Student's t. (Note: this is for a single observation, not $x_1, ..., x_n$.)

The marginal of x is

$$\begin{aligned} p(x) &= \int_\Theta p(x|\sigma^2)p(\sigma^2)\delta\sigma^2 \\ p(x) &= \int_0^\infty (2\pi\sigma^2)^{-1/2} e^{-\frac{1}{2\sigma^2}x^2} * \frac{b^a}{\Gamma(a)}(\sigma^2)^{-a-1}e^{-b/\sigma^2} \\ &= 2^{-1/2}\pi^{-1/2}\frac{b^a}{\Gamma(a)}\int_0^\infty (\sigma^2)^{-a-\frac{1}{2}-1}e^{-\frac{1}{\sigma^2}[\frac{x^2}{2}+b]}\delta\sigma^2 \end{aligned}$$

The integral has the form of the Inverse Gamma pdf for $IG(a + \frac{1}{2}, \frac{x^2}{2} - b)$. This integral is missing the constant of proportionality. If $1 = c * int$, then $int = 1/c$. So the integral term is equal to $\Gamma(a + \frac{1}{2})$. Plug this in to get the form of the Student's t distribution.

$$\frac{1}{\sqrt{2\pi}}\frac{b^a}{\Gamma(a)}\Gamma(a + \frac{1}{2})(\frac{x^2}{2} + b)^{-(a+\frac{1}{2})}$$

Note:

This proof is far easier and nicer if you start with prior $IG(a/2, b/2)$:

$$\begin{aligned} x|\omega &\sim N(\mu, \omega^{-1}) \\ \omega &\sim IG(\frac{d}{2}, d\tau^2 2) \end{aligned}$$

Then the marginal of x is $t(center = \mu, scale = \tau, df = d)$.

The Multivariate Normal Distribution

Basics

Part A

In matrix notation, $cov(x) = E\left\{(x-\mu)(x-\mu)^T\right\}$ where μ is the mean vector whose ith component is $E(x_i)$.

Prove $cov(x) = E(xx^T) - \mu\mu^T$

Begin with the definition of covariance.

$$\begin{aligned}
cov(x) &= E\left\{(x-\mu)(x-\mu)^T\right\} \\
&= E\left\{(x-\mu)(x^T-\mu^T)\right\} \\
&= E\left(xx^T - 2x\mu^T + \mu\mu^T\right) \\
&= E(xx^T) - E(2x\mu^T) + E(\mu\mu^T), \text{ by linearity of expectations} \\
&= E(xx^T) - 2\mu^T E(x) + \mu\mu^T, \text{ since} E(c) = c \text{ and } E(cx) = cE(x) \text{ for constant c} \\
&= E(xx^T) - 2\mu\mu^T + \mu\mu^T, \text{ since } E(x) = \mu \\
&= E(xx^T) - \mu\mu^T
\end{aligned}$$

Prove $cov(Ax+b) = Acov(x)A^T$ for matrix A and vector b

Begin with the definition of covariance.

$$\begin{aligned}
cov(Ax+b) &= E\left\{\left[(Ax+b) - E(Ax+b)\right]\left[(Ax+b) - E(Ax+b)\right]^T\right\} \\
&= E\left\{\left[Ax+b-AE(x)-b\right]\left[Ax+b-AE(x)-b\right]^T\right\} \\
&= E\left\{(Ax-A\mu)(Ax-A\mu)^T\right\}, \text{ since } E(x) = \mu \text{ and the bs cancel} \\
&= E\left\{(Ax-A\mu)(x^TA^T-\mu^TA^T)\right\}, \text{ by distributing the transpose} \\
&= E\left\{A(x-\mu)(x^T-\mu^T)A^T\right\}, \text{ by pulling out } A \text{ and } A^T \\
&= E\left\{A(x-\mu)(x-\mu)^TA^T\right\} \\
&= AE\left\{(x-\mu)(x-\mu)^T\right\}A^T \text{by pulling constants out of the expectation} \\
&= Acov(x)A^T
\end{aligned}$$

Part B

Let z be a random vector $z = (z_1, ..., z_p)^T$, with iid $z_i \sim N(0,1)$. We say z has a standard normal multivariate distribution. Derive the pdf and mgf of z, in vector notation.

Pdf of Multivariate Normal z

Since z_i are independent, the joint pdf is the product of the individual pdfs.

$$p(z) = \prod_{i=1}^{z} (2\pi)^{-1/2} e^{-z_i^2/2} = (2\pi)^{-p/2} exp\left\{\frac{-\sum_{i=1}^{n} z_i^2}{2}\right\}$$

In vector form, $\sum_{i=1}^{n} z_i^2 = z^T z$, so we can rewrite the pdf in vector notation.

$$p(z) = (2\pi)^{-p/2} exp\left\{-\frac{1}{2} z^T z\right\}$$

Mgf of z

The definition of the mgf of a random variable vector is $M_x(t) = E(e^{t^T x})$ in vector notation (pg 3, note 5).

Let $z = (z_1, ..., z_p)^T \sim iidN(0,1)$. We know the standard normal univariate mgf is $E[e^{\frac{1}{2}t^2}]$.

The mgf of z is

$$\begin{aligned} M_z(t) &= E[e^{t^T z}] \\ &= E[e^{\sum_{i=1}^{n} t_i z_i}] \\ &= E[e^{(t_1 z_1)} \cdots e^{(t_n z_n)}] \text{ since z's iid} \\ &= E[e^{(t_1 z_1)}] \cdots E[e^{(t_n z_n)}] \text{ since E(product) = product(E) for iid} \end{aligned}$$

Each of these is the mgf of a univariate standard normal, which has form $E[e^{\frac{1}{2}t^2}]$.

$$\begin{aligned} &= \left(e^{\frac{1}{2}t_1^2}\right) \cdots \left(e^{\frac{1}{2}t_n^2}\right) \\ &= e^{\frac{1}{2}t^T t} \end{aligned}$$

This is the mgf of the standard normal multivariate distribution.

Part C

Prove that $X \sim N(\mu, \Sigma)$ iff its mgf has form $E\left(e^{t^T x}\right) = exp\left\{t^T\mu + t^T\Sigma t/2\right\}$.

(Direction $\rightarrow$)

Let $x = (x_1, ..., x_p)^T$, ie $x \in \mathbb{R}^p$. Let $x \sim N(\mu, \Sigma)$. Want to derive the mgf of x.

Knowns:

$$z = a^T x \sim \text{ Normal } \forall a \in \mathbb{R}^p \text{ (Part B)}$$
$$M_X(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2} \text{ for univariate normal mgf}$$

Plug in $z = a^T x$ to definition of univariate normal mgf.

$$\begin{aligned} M_Z(t) &= E[e^{t^T z}] \\ &= E[e^{ta^T x}] \\ &= E[e^{ta^T x}] \\ &= e^{(mt + \frac{1}{2}vt^2)}, \text{ where } m = E(z) \text{ and } v = var(z) \end{aligned}$$

Then define m and v. Let $E[z] = \mu$, and let $var[z] = \Sigma$.

$$m = E[z] = E[a^T x] = a^T E[x] = a^T \mu$$
$$v = Var[z] = a^T var(x) a = a^T \Sigma a$$

Plug in these definitions to obtain form of multivariate normal mgf.

$$M_Z(t) = e^{(a^T\mu t + \frac{1}{2}a^T\Sigma a t^2)}$$

This holds for any t, so let $t = 1$.

$$M_Z(t) = e^{(a^T\mu + \frac{1}{2}a^T\Sigma a)}$$

This is the form of the multivariate normal mgf. It holds for any vector a.

(Direction $\leftarrow$)

Begin wwith the form of the mgf. Suppose $E[e^{a^T x}] = e^{(a^t\mu + \frac{1}{2}a^T\Sigma a)}$.

Define $z = a^T x$. If $t \in \mathbb{R}$, where t is scalar, then

$$\begin{aligned} E[e^{tz}] &= E[e^{ta^T x}] \\ &= e^{(ta^T\mu + \frac{1}{2}t^2 a^T\Sigma a)} \\ &= e^{tm + \frac{1}{2}vt^2} \end{aligned}$$

This is the univariate normal mgf with $m = a^T\mu$ and $v = a^T\Sigma a$.

We know that $x = (x_1, ..., x_p)^T$ is multivariate normal iff every linear combo of its components is univariate normal. Since $z = a^T x$ is a linear combo of components of x, and z is univariate normal, thereforex is multivariate normal.

Part D

Let z have a standard multivariate normal distribution. Define the random vector $x = Lz + \mu$ for (pxp) matrix L of full column rank. Prove that x is multivariate normal.

Let $x = Lz + \mu$ as described above.

Note that the MGF of z is $M_z(t) = exp\left\{\frac{1}{2}t^T t\right\}$, from Part B.

$$\begin{aligned}
M_x(t) &= E(e^{t^T x}), \text{by definition (Part B)} \\
&= E\left(e^{t^T(Lz+\mu)}\right), \text{by subbing in definition of x} \\
&= E\left(e^{t^T Lz} e^{t^T \mu}\right) \\
&= e^{t^T \mu} E\left(e^{(L^T t)^T z}\right), \text{since } e^{t^T \mu} \text{ doesn't depend on z} \\
&= exp\left\{t^T \mu + \frac{tLL^T t}{2}\right\}
\end{aligned}$$

since $E\left(e^{(L^T t)^T z}\right)$ has the form of $M_z(s) = exp\left\{\frac{1}{2}s^T s\right\}$ from B (std mvn mgf).

Therefore, the mgf for the multivariate normal distribution $x \sim N(\mu, \Sigma = LL^T)$ is

$$M_X(t) = exp\left\{t^T \mu + \frac{tLL^T t}{2}\right\}$$

Part E

Let $X \sim N(\mu, \Sigma)$ be a multivariate normal random variable. Prove X can be written as an affine transformation ($X = LZ + \mu$) of iid standard normal random variables $Z = (z_1, ..., z_n)^T$. Let L be some non-singular matrix. We can then write $Z = L^{-1}(X - \mu)$.

From previous sections, $M_X(t) = E\left(e^{t^T x}\right) = exp\left\{t^T\mu + \frac{t^T\Sigma t}{2}\right\}$.

Since Σ is positive semi-definite, we can write $\Sigma = LL^T$.

Then the mgf of random variable Z is as follows.

$$\begin{aligned} M_Z(t) &= E[e^{t^T z}] = E[e^{t^T L^{-1}(x-\mu)}], \text{ by subbing in } Z = L^{-1}(X-\mu) \\ &= E[e^{t^T L^{-1} x}]e^{-t^T L^{-1}\mu} \\ &= E[e^{(L^{-T}t)^T x}]e^{-t^T L^{-1}\mu} \\ &= E[e^{(L^{-T}t)^T x}]e^{-t^T L^{-1}\mu} \end{aligned}$$

The first term has the form of the multivariate normal mgf. Sub in the definition from above.

$$\begin{aligned} &= exp\left[(L^{-T}t)^T\mu + \frac{(L^{-T}t)^T LL^T L^{-T}t}{2}\right] \\ &= exp\left[-t^T L^{-1}\mu\right] exp\left[\frac{(L^{-T}t)^T LL^T L^{-T}t}{2}\right], \text{ cancelling terms and distributing transpose} \\ &= e^{\frac{t^T t}{2}} \end{aligned}$$

This is the form of the standard normal mgf. Therefore, $z \sim N(0, I)$. Since $X = Lz + \mu$, x is a linear combination of standard normals.

For an algorithm to simulate multivariate normal random variables with a specified mean and covariance matrix:

1. Generate n standard normal univariate random variables z.
2. Let μ be the vector of desired means.
3. Let LL^T be the desired covariance matrix.
4. Construct the multivariate normal distribution using $X = Lz + \mu$.

Regarding decomposition of Σ:

- Using $\Sigma = LL^T$ is Cholesky decomposition.
- Can also use Spectral Decomposition (eigenvalue decomp) of Σ:

Cholesky is 3x faster, but may be less stable if one or more of eigenvalues is tiny. See R snippet below for how these work in terms of decomposing and obtaining L.

```r
#1. Cholesky:
t(chol(Sigma)) %*% chol(Sigma)  #equals Sigma
L = t(chol(Sigma))              #Assign L so LL^T = Sigma
L %*% t(L)                      #Verify LL^T = Sigma

#2. Spectral Decomposition: A = V diag(lambda) V^(-1)
eg = eigen(Sigma)               #Store spectoral value decomposition of Sigma.
V = eg$vectors                  #Extract eigen vectors.
lam = diag(eg$values)               #Extract diagonal matrix of eigenvalues.
V %*% lam %*% solve(V)          #Check reproducing Sigma.

L = V %*% sqrt(lam)             #Assign L so LL^T = Sigma
L %*% t(L)                      #Verify LL^T = Sigma
```

Part F

Use the previous result and the standard normal multivariate pdf to show that the pdf of $X \sim N(\mu, \Sigma)$ has the form $p(x) = Cexp[-\frac{1}{2}Q(x-\mu)]$ for C constant and quadratic form $Q(x-\mu)$.

For $Z \sim N(0, I)$,

$$f(z) = (2\pi)^{-\frac{p}{2}} e^{-\frac{1}{2}z^T z}$$

From Part E,

$$X = LZ + \mu \sim N(\mu, \Sigma), \text{ letting } \Sigma = LL^T$$

Use the transformation theorem,

$$f_Y(y) = f_X\left(g^{-1}(y)\right)|J|$$

1) Since $X = LZ + \mu$,

$$Z = L^{-1}(X - \mu)$$
$$g^{-1}(x) = L^{-1}(x - \mu)$$

2) Since L is a non-singular matrix, the transformation is 1-1.

3) The Jacobian is $J = L^{-1}$. (7) Then

$$|J| = det(L^{-1}) = det(\Sigma^{-1/2})^{-1} = |\Sigma|^{-1/2}$$

4) Plug into the transformation formula.

$$\begin{aligned} f_X(x) &= (2\pi)^{-\frac{p}{2}}|\Sigma|^{-1/2} exp\left\{-\frac{1}{2}[L^{-1}(x-\mu)]^T[L^{-1}(x-\mu)]\right\} \\ &= (2\pi)^{-\frac{p}{2}}|\Sigma|^{-1/2} exp\left\{-\frac{1}{2}(x-\mu)^T L^{-T} L^{-1}(x-\mu)\right\} \\ &= (2\pi)^{-\frac{p}{2}}|\Sigma|^{-1/2} exp\left\{-\frac{1}{2}(x-\mu)^T (LL^T)^{-1/2}(x-\mu)\right\} \end{aligned}$$

which has the desired form.

Part G

Let $x_1 \sim N(\mu_1, \Sigma_1)$ and $x_2 \sim N(\mu_2, \Sigma_2)$ where $x_1 \perp x_2$. Let $y = Ax_1 + Bx_2$ for A_1, B of full rank and appropriate dimension. Use previous results to characterize the distribution of y.

Begin with definition of mgf of y.

$$\begin{aligned}
M_Y(t) &= E[e^{t^T y}] \\
&= E[e^{t^T(Ax_1+Bx_2)}] \\
&= E[e^{t^T Ax_1} e^{t^T Bx_2}] \\
&= E[e^{t^T Ax_1}]E[e^{t^T Bx_2}], \text{ by definition} \\
&= E[e^{(A^T t)^T x_1}]E[e^{(B^T t)^T x_2}] \\
&= e^{\left[(A^T t)^T \mu_1 + \frac{1}{2}(A^T t)^T \Sigma_1 (A^T t)\right]} * e^{\left[(B^T t)^T \mu_2 + \frac{1}{2}(B^T t)^T \Sigma_2 (B^T t)\right]}, \text{ by mvn mgf definition} \\
&= e^{\left[(A^T t)^T \mu_1 + (B^T t)^T \mu_2 + \frac{1}{2}\left((A^T t)^T \Sigma_1 (A^T t) + (B^T t)^T \Sigma_2 (B^T t)\right)\right]} \\
&= e^{\left[t^T(A\mu_1 + B\mu_2) + \frac{1}{2} t^T \left(A\Sigma_1 A^T + B\Sigma_2 B^T\right) t\right]}
\end{aligned}$$

This has the form of the multivariate normal mgf. Therefore

$$y = Ax_1 + Bx_2 \sim N(A\mu_1 + B\mu_2, A\Sigma_1 A^T + B\Sigma_2 B^T)$$

Conditionals and Marginals

Part A

Let $X \sim N(\mu, \Sigma)$. Let $x = (x_1, x_2)^T$ be an arbitrary partition of x into two components, of lengths k and $q = p - k$ respectively. Partition $\mu = (\mu_1, \mu_2)^T$ and $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ where $\Sigma_{12} = \Sigma_{21}^T$. Derive the marginal distribution of x_1. (Use affine transform result.)

Let A be a matrix, so that $A = [I_{(kxk)}|0_{(kxq)}]$. Note: | indicates concatenated (cbind).

Then $x_1 = A^T x$. By our earlier result (Part G),

$$y = Ax_1 + Bx_2 \sim N\left(A\mu_1 + B\mu_2, A\Sigma_1 A^T + B\Sigma_2 B^T\right)$$

Therefore, $x_1 \sim N(\mu_1, \Sigma_{11})$.

Illustration:

Say $x = [x_1, ..., x_6]^T$ and we want the marginal of the first two elements, $x_1 = [x_1, x_2]$.

Mean:

$$Ax = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Cov:

$$A\Sigma A^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} & v_{16} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} & v_{26} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} & v_{36} \\ v_{41} & v_{42} & v_{43} & v_{44} & v_{45} & v_{46} \\ v_{51} & v_{52} & v_{53} & v_{54} & v_{55} & v_{56} \\ v_{61} & v_{62} & v_{63} & v_{64} & v_{65} & v_{66} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} & v_{16} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} & v_{26} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} = \Sigma_{11}$$

Part B

Let $\Omega = \Sigma^{-1}$ be the precision matrix, ie the inverse covariance matrix. It is usually easier for multivariate normals to work with precision instead of covariance.

Partition Ω as we did Σ, so $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ and $\Omega = \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix}$. Derive blocks of Ω in terms of blocks of Σ.

We know $\Sigma\Sigma^{-1} = I_{nxn} \rightarrow \Sigma\Omega = I_{nxn}$ by definition of identities. So we can set up the equation

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix} = \begin{pmatrix} I_p & 0 \\ 0 & I_q \end{pmatrix}$$

This yields a system of four equations.

$$\Sigma_{11}\Omega_{11} + \Sigma_{12}\Omega_{21} = I_p \tag{1}$$
$$\Sigma_{11}\Omega_{12} + \Sigma_{12}\Omega_{22} = 0 \tag{2}$$
$$\Sigma_{21}\Omega_{11} + \Sigma_{22}\Omega_{21} = 0 \tag{3}$$
$$\Sigma_{21}\Omega_{12} + \Sigma_{22}\Omega_{22} = I_q \tag{4}$$

Solving (2) and (3) yields:

$$\Omega_{12} = -\Sigma_{11}^{-1}\Sigma_{12}\Omega_{22} \tag{5}$$
$$\Omega_{21} = -\Sigma_{22}^{-1}\Sigma_{21}\Omega_{11} \tag{6}$$

Then plug the (6) result into (1), and the (5) result into (4), to obtain

$$\Sigma_{11}\Omega_{11} + \Sigma_{12}\Omega_{21} = I_p \rightarrow \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)\Omega_{11} = I_p \rightarrow \Omega_{11} = \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}$$
$$\Sigma_{21}\Omega_{12} + \Sigma_{22}\Omega_{22} = I_q \rightarrow \left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)\Omega_{22} = I_q \rightarrow \Omega_{22} = \left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)^{-1}$$

Plug these results into (5) and (6) to obtain

$$\Omega_{12} = -\Sigma_{11}^{-1}\Sigma_{12}\Omega_{22} \rightarrow \Omega_{12} = -\Sigma_{11}^{-1}\Sigma_{12}\left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)^{-1}$$
$$\Omega_{21} = -\Sigma_{22}^{-1}\Sigma_{21}\Omega_{11} \rightarrow \Omega_{21} = -\Sigma_{22}^{-1}\Sigma_{21}\left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}$$

Therefore:

$$\Omega_{11} = \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}$$
$$\Omega_{22} = \left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)^{-1}$$
$$\Omega_{12} = -\Sigma_{11}^{-1}\Sigma_{12}\left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)^{-1}$$
$$\Omega_{21} = -\Sigma_{22}^{-1}\Sigma_{21}\left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}$$

Part C

Derive the multivariate normal conditional density of $f(x_1|x_2)$, given the same partitioning as in Part A and Part B.

Begin with a few givens:

$$f(x) = f(x_1, x_2) \sim N(\mu, \Sigma)$$
$$f(x_2) \sim N(\mu_2, \Sigma_{22}) \text{ is the marginal for } x_2, \text{ from above}$$
$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)}$$

Therefore,

$$f(x_1, x_2) \propto \exp[-\frac{1}{2}(x-\mu)^T\Omega(x-\mu)] \text{where} \Omega = \Sigma^{-1} \text{ from Part B}$$
$$f(x_2) \propto \exp[-\frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2)]$$

The conditional density therefore has the form

$$f(x_1|x_2) \propto \exp[-\frac{1}{2}Q_1(x_1, x_2) + \frac{1}{2}Q_2(x_2)] = \exp[-\frac{1}{2}\left(Q_1(x_1, x_2) - Q_2(x_2)\right)]$$

We will work with the density in the log form, beginning with $Q_1(x_1, x_2) - Q_2(x_2)$. First, rewrite $Q_1(x_1, x_2)$ in terms of the blocks of $x^{(**)}$.

$$\begin{aligned}
&(x-\mu)^T\Omega(x-\mu) \\
&= \begin{bmatrix} (x_1-\mu_1)^T & (x_2-\mu_2)^T \end{bmatrix} \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} (x_1-\mu_1) \\ (x_2-\mu_2) \end{bmatrix} \\
&= (x_1-\mu_1)^T\Omega_{11}(x_1-\mu_1) + (x_2-\mu_2)^T\Omega_{21}(x_1-\mu_1) + (x_1-\mu_1)^T\Omega_{12}(x_2-\mu_2) + (x_2-\mu_2)^T\Omega_{22}(x_2-\mu_2) \\
&= (x_1-\mu_1)^T\Omega_{11}(x_1-\mu_1) + 2(x_1-\mu_1)^T\Omega_{12}(x_2-\mu_2) + (x_2-\mu_2)^T\Omega_{22}(x_2-\mu_2), \text{ since } \Omega_{21} = \Omega_{12}^T
\end{aligned}$$

$(**)$ This has form $\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} c & d \\ e & f \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = aca + bea + adb + bfb$

Put this back together with the $Q_2(x_2)$ term to get the entire form of the exponential term:

$$= (x_1-\mu_1)^T\Omega_{11}(x_1-\mu_1) + 2(x_1-\mu_1)^T\Omega_{12}(x_2-\mu_2) + (x_2-\mu_2)^T\Omega_{22}(x_2-\mu_2) + (x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2)$$

The last two terms are constant with respect to x_1, so we can ignore these and focus on the first two terms:

$$(x_1-\mu_1)^T\Omega_{11}(x_1-\mu_1) + 2(x_1-\mu_1)^T\Omega_{12}(x_2-\mu_2)$$

For ease in calculations and without loss of generality, assume for now $\mu_1 = \mu_2 = 0$. We will add them back later.

$$\begin{aligned}
&= x_1^T\Omega_{11}x_1 + 2x_1^T\Omega_{12}x_2 \\
&= x_1^T\Omega_{11}x_1 + x_2^T\Omega_{12}^Tx_1
\end{aligned}$$

This is a quadratic form, $x^TCx + 2b^Tx$. Note, we are ignoring constants which do not matter here.

Completing the square will yield the form $(x+m)^T M(x+m)$, where $M = C$ and $m = -C^{-1}b$.

$$\begin{aligned} M &= C = \Omega_{11} \\ b &= (x_2^T \Omega_{12}^T)^T = \Omega_{12} x_2 \\ m &= -C^{-1}b = -\Omega_{11}^{-1}\Omega_{12}x_2 \end{aligned}$$

Plugging in to the quadratic form, we get

$$= (x_1 - \Omega_{11}^{-1}\Omega_{12}x_2)^T \Omega_{11}(x_1 - \Omega_{11}^{-1}\Omega_{12}x_2)$$

Now we can plug the means back in; replace x_1 and x_2 with $x_1 - \mu_1$ and $x_2 - \mu_2$.

$$\begin{aligned} &= \left[(x_1-\mu_1) - \Omega_{11}^{-1}\Omega_{12}(x_2-\mu_2)\right]^T \Omega_{11} \left[(x_1-\mu_1) - \Omega_{11}^{-1}\Omega_{12}(x_2-\mu_2)\right] \\ &= \left[x_1 - \left(\mu_1 + \Omega_{11}^{-1}\Omega_{12}(x_2-\mu_2)\right)\right]^T \Omega_{11} \left[x_1 - \left(\mu_1 + \Omega_{11}^{-1}\Omega_{12}(x_2-\mu_2)\right)\right] \end{aligned}$$

This is the form of the multivariate normal; just need to convert Ω back to Σ.

$$\begin{aligned} \Omega_{11} &= (\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} \\ \Omega_{11}^{-1}\Omega_{12} &= \Sigma_{12}\Sigma_{22}^{-1} \end{aligned}$$

Therefore, the conditional distribution of $f(x_1|x_2)$ is multivariate normal, with parameters

$$\begin{aligned} \mu_{x_1|x_2} &= \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ \Sigma_{x_1|x_2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned}$$

NOTE 1: Simplification of $\Omega_{11}^{-1}\Omega_{12}$

$$\begin{aligned} &\Omega_{21} = \Omega_{12}^T = -\Sigma_{22}^{-1}\Sigma_{12}^T\Omega_{11}, \text{ from Part B} \\ &\Omega_{11}^{-1}\Omega_{12} = ((\Omega_{11}^{-1}\Omega_{12})^T)^T = (-\Sigma_{22}^{-1}\Sigma_{12}^T)^T = -\Sigma_{12}\Sigma_{22}^{-1} \end{aligned}$$

Derivation of Multivariate Completing the Square

Begin with quadratic equation $x'Ax - 2b'x$	(7)
Want to end with $(x-c)'A(x-c)$	(8)
Expand desired end result to get $x'Ax - 2c'Ax + c'Ac$. Can ignore last constant term.	(9)
Match up equations (1) and (2) to see that $A = A, c'A = b' \rightarrow A'c = b \rightarrow c' = A^{-1}b$	(10)

Multiple regression: three classical principles for inference

Consider linear model $y_i = x_i^T\beta + \epsilon_i$ for $i = 1, ..., n$. y_i is a scalar response, x_i is a p-vector of predictors/features, ϵ_i are the errors. $\hat{\beta}$ denotes the esstimate for β.

Part A:

Show that least squares, maximumm likelihood under Gaussianity, and method of moments result in the same estimate of $\hat{\beta}$.

Least Squares:

$$\begin{aligned}\hat{\beta} &= \underset{\beta \in R^p}{\text{argmin}} \left\{ \sum_{i=1}^{n} (y_i - x_i\beta)^2 \right\} \\ &= \underset{\beta \in R^p}{\text{argmin}} \left\{ (y - X\beta)^T (y - X\beta) \right\}\end{aligned}$$

To minimize $\hat{\beta}$, take the derivative of the objective function wrt β and set equal to zero. Solve for β to find the $\hat{\beta}$ which minimizes.

$$\begin{aligned}&= (y - X\beta)^T (y - X\beta) \\ &= (y^T - \beta^T X^T)(y - X\beta) \\ &= y^T y - 2y^T X\beta + \beta^T X^T X\beta\end{aligned}$$

Take the derivative with respect to β and set equal to zero.

$$\begin{aligned}&\frac{\delta}{\delta\beta} \left[y^T y - 2y^T X\beta + \beta^T X^T X\beta \right] = 0 \\ &0 - 2y^T X + 2X^T X\beta = 0 \\ &X^T X\hat{\beta} = y^T X \\ &\hat{\beta} = (X^T X)^{-1} X^T y\end{aligned}$$

The Least-Squares estimate of β is $\hat{\beta} = (X^T X)^{-1} X^T y$

Maximum Likelihood:

$$\hat{\beta} = \underset{\beta \in R^p}{\text{argmax}} \left\{ \prod_{i=1}^{n} p(y_i | \beta, \sigma^2) \right\}$$

Model is

$$y_i = x_i\beta + \epsilon_i, \text{ and } \epsilon_i \sim iidN(0, \sigma^2), \text{ where } \epsilon_i = y_i - x_i^T\beta$$

Likelihood is

$$\begin{aligned}p(y_i | \beta, \sigma^2) &\propto \exp[-\frac{1}{2}(y_i - x_i^T\beta)^2] \\ &= \exp[-\frac{1}{2}(y - X\beta)^T \frac{1}{\sigma^2} I (y - X\beta)]\end{aligned}$$

We can maximize the log, since the log is monotonically increasing. Taking the log gives

$$-\frac{1}{2}(y-X\beta)^T\frac{1}{\sigma^2}I(y-X\beta)]$$

Pull out constant σ^2 and get rid of I.

$$=-\frac{1}{2\sigma^2}(y-X\beta)^T(y-X\beta)$$

To maximize this, we can take the derivative wrt β set it equal to zero and solve for β. The constant $-\frac{1}{2\sigma^2}$ drops out, we multiply the terms of the product, and we are left with the same maximization problem as we had previously.

$$\frac{\delta}{\delta\beta}\left[y^Ty-2y^TX\beta+\beta^TX^TX\beta\right]=0$$

This yields the same estimator: $\hat{\beta}=(X^TX)^{-1}X^Ty$

Method of Moments:

Choose $\hat{\beta}$ so that the sample covariance between the errors and each of the predictors is exactly zero. Let x_j indicate each predictor, ie columns of design matrix X.

$$\begin{aligned}
&c\hat{o}v(\epsilon,x_j)=0\\
&\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(e_i-\bar{e})=0\\
&\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}e_i)-\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}\bar{e})-\frac{1}{n-1}n\bar{x}_je_i+\frac{1}{n-1}n\bar{x}_j\bar{e}=0\\
&\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}e_i)-\frac{1}{n-1}n\bar{x}_j\bar{e}-\frac{1}{n-1}n\bar{x}_je_i+\frac{n}{n-1}\bar{x}_j\bar{e}=0
\end{aligned}$$

We can mean-center the data without loss of generality, so $\bar{x}_j=0\;\forall j\in(1...p)$.

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}e_i)=0\\
&X^Te=0
\end{aligned}$$

Since $y=X\beta+e$, then $e=y-X\beta$.

$$\begin{aligned}
&X^T(y-X\beta)=0\\
&X^Ty-X^TX\beta=0\\
&\hat{\beta}=(X^TX)^{-1}X^Ty
\end{aligned}$$

Part B:

Derive β estimator:

Find $\hat{\beta}$ estimate for weighted least squares, by minimizing the weighted sum of squared errors.

$$\begin{aligned}\hat{\beta} &= \underset{\beta \in R^p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} w_i (y_i - x_i \beta)^2 \right\} \\ &= \underset{\beta \in R^p}{\operatorname{argmin}} \left\{ (y - X\beta)^T W (y - X\beta) \right\}\end{aligned}$$

where W is a diagonal matrix containing weights w_i on the diagonal.

To minimize $\hat{\beta}$, take the derivative of the objective function wrt β and set equal to zero. Solve for β to find the $\hat{\beta}$ which minimizes.

$$\begin{aligned}&= (y - X\beta)^T W (y - X\beta) \\ &= (y^T - \beta^T X^T) W (y - X\beta) \\ &= y^T W y - 2 y^T W X \beta + \beta^T X^T W X \beta\end{aligned}$$

Take the derivative with respect to β and set equal to zero.

$$\begin{aligned}&\frac{\delta}{\delta \beta} \left[y^T W y - 2 y^T W X \beta + \beta^T X^T W X \beta \right] = 0 \\ &0 - 2 y^T W X + 2 X^T W X \beta = 0 \\ &X^T W X \hat{\beta} = y^T W X \\ &\hat{\beta} = (X^T W X)^{-1} X^T W y\end{aligned}$$

Compare to maximum-lhood solution under heteroscedastic Gaussian error:

Errors are $\epsilon_i \sim N(0, \sigma_i^2)$ where $y_i = x_i \beta + \epsilon_i \rightarrow \epsilon_i = (y_i - x_i \beta)$

The likelihood is

$$\prod_{i=1}^{n} p(y_i | \beta, \sigma_i^2) \propto \exp[-\sum_{i=1}^{n} \frac{1}{2\sigma_i^2} (y_i - x_i \beta)^2]$$

Let W be a diagonal matrix, with diagonal elements $\frac{1}{\sigma_i^2}$. Then the likelihood is

$$= \exp[-\frac{1}{2} (y - X\beta)^T W (y - X\beta)]$$

Minimize in the same way as Part A of this section, obtaining the same estimator $\hat{\beta} = (X^T W X)^{-1} X^T W y$.

The key is that the weights w_i are the precisions, so $w_i = \frac{1}{\sigma_i^2}$.

Quantifying Uncertainty: Some basic frequentist ideas

In Linear Regression
Suppose data observed from a linear model with Gaussian error: $y = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2 I)$.

Part A

Derive the sampling distribution of the estimator for β from the previous problem, $\hat{\beta} = (X^T X)^{-1} X^T y$.

First, $\epsilon = y - X\beta$ is multivariate normal. Therefore, $y \sim N(X\beta, \sigma^2 I)$, since this is shifting a multivariate normal by a constant.

Then $\hat{\beta} = (X^T X)^{-1} X^T y$ is a linear combination of Y. Can say $\hat{\beta} = Ay$ where $A = (X^T X)^{-1} X^T$.

Then based on the results from the multivariate normal section,

$$
\begin{aligned}
&E[\hat{\beta}] = E[Ay] = AE[y] = AX\beta = (X^T X)^{-1} X^T X\beta = \beta \\
&Cov(\hat{\beta}) = Cov(Ay) = ACov(y)A^T \\
&= (X^T X)^{-1} X^T [\sigma^2 I] X (X^T X)^{-1} \\
&= \sigma^2 (X^T X)^{-1} X^T X (X^T X)^{-1} \\
&= \sigma^2 (X^T X)^{-1}
\end{aligned}
$$

The sampling distribution of $\hat{\beta} \sim N\left(\beta, \sigma^2 (X^T X)^{-1}\right)$.

Part B

Propose a strategy for calculating the standard errors of each $\hat{\beta}_j$.

General strategy

To estimate the standard errors, we can follow the usual strategy of using the sampling distribution variance derived in the previous part, and plug in an estimated value of σ^2. Take the square root of the result to obtain a standard error estimate.

$$\hat{Var}\hat{\beta}_j = \hat{\sigma}^2(X^TX)^{-1}$$

Estimating σ^2

σ^2 measures how much each individual y_i response varies around the unknown population regression line. We can estimate this by measuring how much our observed y_i values vary around the estimated population regression line.

We can estimate this variation using the mean squared error, $MSE = \frac{RSS}{n-p} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p}$ where
n = number obs, p = number predictors.

This gives an average of the squared error between the true observation and the observation as predicted by the model.

Proof: Why MSE is an unbiased estimator of σ^2

$$RSS = ||y - \hat{y}||_2^2$$
$$E(RSS) = E[\ ||y - X(X'X)^{-1}X'y||_2^2$$

Call $H = X(X'X)^{-1}X'$, so have $E[\ ||y - Hy||_2^2\ ]$, and note that H is the perpendicular projection matrix onto the column space of X, $C(X)$.

- H is the ppm onto $C(X)$, so
- $(I - H)$ is the ppm onto the orthogonal complement of $C(X)$, $C(X)^{\perp}$
- Then $(I - H)X = 0$, and (I-H) is symmetric and idempotent.

$$\begin{aligned}
&E[\ ||y - Hy||_2^2\ ] \\
&\quad = E[\ ||(I - H)y||_2^2\ ] \\
&\quad = E[\ ||(I - H)(X\beta + e)||_2^2\ ] \\
&\quad = E[\ ||(I - H)e||_2^2\ ] \\
&\quad = E[e'(I - H)(I - h)e] = E[e'(I - H)e]
\end{aligned}$$

Then use the "trace trick" (a scalar = trace of the scalar), can write:

$$
\begin{aligned}
&= E[tr(e'(I-H)(I-H)e], \text{ then cyclic permute inside trace} \\
&= E[tr((I-H)(I-H)ee')] \\
&= E[tr((I-H)ee'] \\
&= tr(I-H)E[ee'] \\
&= tr(I-H)\sigma^2 I \\
&= \sigma^2 tr(I-H) \\
&= \sigma^2 \left(tr(I_{nxn}) - tr(H)\right) \\
&= \sigma^2 \left(tr(I_{nxn} - tr(X(X'X)^{-1}X')\right), \text{ then cyclic permute inside trace} \\
&= \sigma^2 \left(tr(I_{nxn} - tr((X'X)^{-1}X'x)\right) \\
&= \sigma^2 \left(tr(I_{nxn} - tr(I_{pxp})\right) \\
&= \sigma^2 (n-p)
\end{aligned}
$$

Therefore, $E[\frac{RSS}{n-p}] = \sigma^2$, and so $MSE = \frac{RSS}{n-p}$ is an unbiased estimator of σ^2.

R Results

```
# My Results:
           V5     V6     V7     V8     V9    V10    V11    V12    V13
38.329  0.007  0.174  0.024  0.069  0.125  0.000  0.015  0.119  0.005

# LM Results:
           V5     V6     V7     V8     V9    V10    V11    V12    V13
38.329  0.007  0.174  0.024  0.069  0.125  0.000  0.015  0.119  0.005
```

See R Appendix for code.

Propogating Uncertainty
Say $\theta = (\theta_1, ..., \theta_p)^T$ are from a multivate model, perhaps a regression model. Estimates are $\hat{\theta} = (\hat{\theta_1}, ..., \hat{\theta_p})^T$. To resport uncertainty about the $\hat{\theta}_j$s, can use the diagonal of the estimated covariance matrix: $\hat{\Sigma}_{jj} = \hat{\sigma}_j^2$.

Estimated covariance matrix is $\hat{\Sigma} = cov(\hat{\theta}) = E\left[(\hat{\theta} - \bar{\theta})(\hat{\theta} - \bar{\theta})^T\right]$.

Part A

Want to estimate $f(\theta_1 + \theta_2)$. Calculate the standard error of $f(\hat{\theta})$ and generalize to the case where f is sum of all p components of θ.

From previous sections, $\hat{\theta}$ is normally distributed. So $\hat{\theta}_1 + \hat{\theta}_2$ is a linear combinations of normals.

$$
\begin{aligned}
var(x + y) &= var(x) + var(y) + 2cov(x, y) \\
var(\hat{\theta}_1 + \hat{\theta}_2) &= var(\hat{\theta}_1) + var(\hat{\theta}_2) + 2cov(\hat{\theta}_1, \hat{\theta}_2) \\
var(\hat{\theta}_1 + \hat{\theta}_2) &= \hat{\Sigma}_{11} + \hat{\Sigma}_{22} + 2\hat{\Sigma}_{12}
\end{aligned}
$$

To generalize to a case involving the sum of the p θ_js:

$$
var(\sum_{i=1}^{p} \hat{\theta}_i) = \left(\sum_{i=1}^{p} \hat{\Sigma}_{ii}\right) + 2\left(\sum_{i=1}^{p} \sum_{j=1_{j \neq i}}^{p} \hat{\Sigma}_{ij}\right)
$$

Part B

How to estimate $f(\hat{\theta})$ when f is a non-linear function of the thetas?

Delta method could work. Can use a Taylor approximation (which the Delta Method is based on).

$$
\begin{aligned}
g(\hat{\theta}) &\approx g(\theta) + g'(\theta)^T(\hat{\theta} - \theta) \\
&= g(\theta) + \sum_{i=1}^{p} g_i'(\hat{\theta}_i - \theta_i)
\end{aligned}
$$

Then plug this first-order Taylor Series approximation into the variance:

$$
var\left(g(\hat{\theta})\right) \approx var\left(g(\theta) + \sum_{i=1}^{p} g_i'(\hat{\theta}_i - \theta_i)\right)
$$

Bootstrapping

Part A

The bootstrap estimate reasonably approximated the normal theory parametric covariance matrix. See R Appendix for code. Output for the bootstrapped covariance matrix versus the parametric estimate:

```
#Display bootstrapped estimate. (B=10000)
> round(mybetacov,2)
         [,1]  [,2]  [,3]  [,4] [,5]  [,6] [,7]  [,8]  [,9] [,10]
 [1,] 1405.35 -0.26 -2.00 -0.14  0.3  1.60    0 -0.03  0.39     0
 [2,]   -0.26  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
 [3,]   -2.00  0.00  0.03  0.00  0.0  0.00    0  0.00  0.00     0
 [4,]   -0.14  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
 [5,]    0.30  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
 [6,]    1.60  0.00  0.00  0.00  0.0  0.02    0  0.00 -0.01     0
 [7,]    0.00  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
 [8,]   -0.03  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
 [9,]    0.39  0.00  0.00  0.00  0.0 -0.01    0  0.00  0.01     0
[10,]    0.00  0.00  0.00  0.00  0.0  0.00    0  0.00  0.00     0
>
>
> #Display the parametric normal theory estimate.
> round(betacovlm,2)
            x   xV5   xV6   xV7  xV8   xV9 xV10  xV11  xV12 xV13
x    1469.09 -0.28 -2.06 -0.15 0.36  1.59    0 -0.04  0.42    0
xV5    -0.28  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
xV6    -2.06  0.00  0.03  0.00 0.00  0.00    0  0.00  0.00    0
xV7    -0.15  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
xV8     0.36  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
xV9     1.59  0.00  0.00  0.00 0.00  0.02    0  0.00 -0.01    0
xV10    0.00  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
xV11   -0.04  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
xV12    0.42  0.00  0.00  0.00 0.00 -0.01    0  0.00  0.01    0
xV13    0.00  0.00  0.00  0.00 0.00  0.00    0  0.00  0.00    0
```

Part B

For a sample size N=1000, I was able to recover the mean and covariance matrix fairly well. See R Appendix for code for each of the three functions.

```
$mu_hat
[1] 3.931601 6.970860

$Sigma_hat
          [,1]     [,2]
[1,] 10.507384 3.444544
[2,]  3.444544 2.255850

> mu           #Output true mu.
[1] 4 7
> Sigma          #Output true Sigma
     [,1] [,2]
[1,]   10    3
[2,]    3    2
```

Part C

The sampling distribution of the multivariate normal MLE parameters look fairly normal.

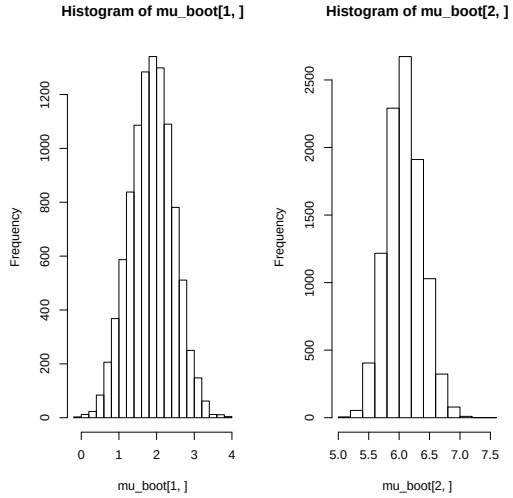

Figure 1: Bootstrapped vector of means

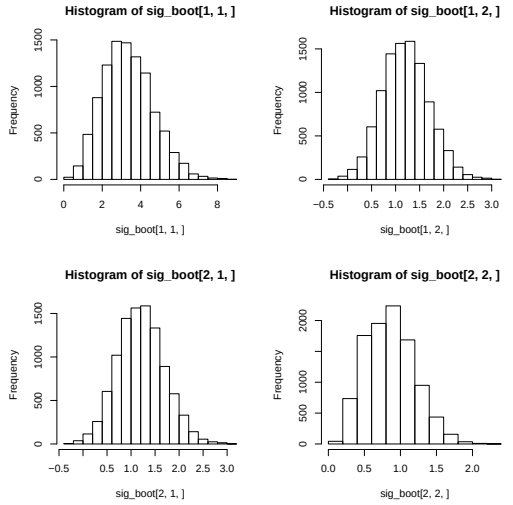

Figure 2: Bootstrapped Cov Matrix Entires

Appendix: R Code

Quantifying Uncertainty - Linear Regression - Part B

```
#SDS 383D. Exercise 01.  January 2016.  Jennifer Starling.

#############################################################
### Quantifying Uncertainty: Some Basic Frequentist Ideas   ###
### Linear Regression                                       ###
### Part B                                                  ###
#############################################################

### PURPOSE:  Linear Regression: Estimate Standard Error of Coefficients Beta

# Load the mlbench library containing the ozone data.
library(mlbench)

# Load the ozone data.
ozone = data(Ozone, package='mlbench')

# Look at the help file for details
?Ozone

# Scrub the missing values and extract the relevant columns
ozone = na.omit(Ozone)[,4:13]

y = ozone[,1]                #Extract response.
x = as.matrix(ozone[,2:10]) #Extract design matrix.
x = cbind(1,x)               #Add intercept to design matrix.

#Compute estimated beta_hat value.
betahat = solve(t(x) %*% x) %*% t(x) %*% y

#Compute sigma2_hat estimate.
yhat = x %*% betahat
rss = sum((y-yhat)^2)
sig2_hat = rss / (length(y) - length(betahat))

#Plug in sigma2_hat to obtain beta cov matrix estimate.
covbeta = sig2_hat * solve(t(x) %*% x)

#Standard error estimate for beta_j by sqrt of diagonals.
se_beta = sqrt(diag(covbeta))

#Compare estimates to those produced by lm method.
# Now compare to lm
lm1 = lm(y~x-1) #The "-1" says not to fit an intercept; we hard-coded it.

summary(lm1)            #Display lm model summary.
betacovlm = vcov(lm1)   #Extract cov matrix from lm model object.
sqrt(diag(betacovlm))   #SE estimates for beta_j from lm object.

#Display comparison of results.
round(se_beta,3)                #my result
round(sqrt(diag(betacovlm)),3)  #lm result
```

Quantifying Uncertainty - Bootstrapping

```
#SDS 383D. Exercise 01.  January 2016.  Jennifer Starling.

#############################################################
### Quantifying Uncertainty: Some Basic Frequentist Ideas   ###
### Bootstrapping                                           ###
### Parts A & B                                             ###
#############################################################

### PURPOSE:  Bootstrap covariance matrix and MLE estimates.

#_____________________________________________________________________
### Bootstrapping Part A

# Let Sigma_hat = cov matrix of sampling dist of beta_hat.
# Write a function that estimates Sigma_hat via bootstrap for a given
# design matrix X and response vector y.  Use it to compute Sigma_hat
# for the ozone data and compare to the parametric theory estimate.

#_____________________
beta_cov_boot = function(X,y,B){
    #OVERVIEW: This function generates B bootstrap realizations of the beta
    # least-squares coefficients.  It then computes the covariance matrix of beta
    # by computing the variances and covariances of each vector of bootstrapped beta_j
        's.

    #INPUTS:    X = nxp design matrix, with no intercept col of 1's (function will add
         these)
    #           y = nx1 response vector
    #           B = number of bootstrap samples.
    #OUTPUTS:   cov_hat = Sigma_hat covariance matrix for betas.

    n = nrow(X)      #Number of observations in design matrix.
    p = ncol(X)      #Number of predictors.

    #Matrix to hold each beta bootstrap sample.  Each row is a sample. Each col is a
        beta_j.
    betahat_boot = matrix(0,nrow=B,ncol=p)

    #Pre-cache (X'X)^-1 X'
    xtx_inv_xt = solve(t(X) %*% X) %*% t(X)

    #Fit model and obtain residuals, e.
    beta_hat = xtx_inv_xt %*% y
    yhat = X %*% beta_hat
    e = y - yhat

    #NOTE: Bootstrapping the residuals only, as we want to treat X as fixed.
    for (b in 1:B){
        samps = sample(1:n,n,replace=T) #Select bootstrap indices.
        e_boot = e[samps]               #Sample residuals.
        y_boot = yhat + e_boot          #Bootstrapped y values.

        #Calculate bootstrapped beta coefficients.
        betahat_boot[b,] = xtx_inv_xt %*% y_boot
    }

    #Estimate cov matrix using var(beta_i,beta_j) for all cols.
```

```
    #(Each col is a vector of B beta_j estimates.)
    beta_hat_cov = matrix(0,nrow=p,ncol=p)

    #Set up list of matrix indices.
    idx = expand.grid(1:p,1:p)

    for (i in 1:10){
        for (j in 1:10){
            #Calculate covar entry.
            beta_hat_cov[i,j] = cov(betahat_boot[,i], betahat_boot[,j])
        } #end j loop
    } #end i loop

    return(beta_hat_cov)
} #END FUNCTION
#------------------------

#Test this out with the Ozone covariance matrix.

# Load the library & data
library(mlbench)
ozone = data(Ozone, package='mlbench')

# Scrub the missing values
# Extract the relevant columns
ozone = na.omit(Ozone)[,4:13]

y = ozone[,1]
x = as.matrix(ozone[,2:10])

# add an intercept
x = cbind(1,x)

#Compute cov matrix using lm.  (-1) means don't fit an intercept; we hard-coded it in
    X.
lm1 = lm(y~x-1)

summary(lm1)
betacovlm = vcov(lm1)
sqrt(diag(betacovlm))

#Run bootstrap function.
mybetacov = beta_cov_boot(x,y,B=10000)

#Display bootstrap estimate.
round(mybetacov,2)


#Display the parametric normal theory estimate.
round(betacovlm,2)

#-------------------------------------------------------------------
### Bootstrapping Part B - 1

mvn_simulate = function(mu,Sigma){
    #PURPOSE: Simulates mvn random variables given a mean mu and cov Sigma.
    #This function returns a single X ~ MVN(mu,Sigma) realization.

    #ALGORITHM:
```

```
    # This algorithm is based on the derivation in Multivariate Normal Part E, which
    # showed that any MVN can be written as a linear combo of standard normals:

    # Simulate Z, a vector of p normal random variables.  (p = desired mvn dimension,
       p = length(mu))
    # Let mu be the vector of means, Sigma be the specified covariance matrix.
    # Let Sigma = LL^T.
    # Simulate mvn rv as X = LZ + mu

    #INPUTS:    mu = desired vector of means.  Must be length p.
    #           Sigma = desired covariance matrix.  Must be (pxp), symmetric, pos
       semidef.
    #OUTPUTS:   x = a realization from MVN(mu,Sigma)

    p = length(mu)      #Set length of mu vector.
    z = rnorm(p,0,1)    #Generate p iid standard normals z_i.

    #Compute L using spectral value decomposition.  V %*% lam %*% solve(V)
    #(See notes below.  Cholesky is 3x faster, spectral is more stable.)

    eg = eigen(Sigma)           #Store spectoral value decomposition of Sigma.
    V = eg$vectors              #Extract eigen vectors.
    lam = diag(eg$values)       #Extract diagonal matrix of eigenvalues.

    L = V %*% sqrt(lam)         #Assign L so LL^T = Sigma

    #Compute realization of x ~ mvn(mu,Sigma)
    x = L %*% z + mu
    return(x)
}

#Test it out:
mu = c(2,5)
Sigma = matrix(c(10,3,3,2),2,2)
x = mvn_simulate(mu,Sigma)

#--------------
#NOTE: Doing LL^T decomposition using eigen() spectral decomposition,
#instead of cholesky.  Cholesky is faster, but eigen is more stable.

#Will show how it is done here both ways.
Sigma <- matrix(c(10,3,3,2),2,2)

#1. Cholesky:
t(chol(Sigma)) %*% chol(Sigma)  #equals Sigma
L = t(chol(Sigma))          #Assign L so LL^T = Sigma
L %*% t(L)                  #Verify LL^T = Sigma

#2. Spectral Decomposition: A = V diag(lambda) V^(-1)
eg = eigen(Sigma)           #Store spectoral value decomposition of Sigma.
V = eg$vectors              #Extract eigen vectors.
lam = diag(eg$values)           #Extract diagonal matrix of eigenvalues.
V %*% lam %*% solve(V)      #Check reproducing Sigma.

L = V %*% sqrt(lam)         #Assign L so LL^T = Sigma
L %*% t(L)                  #Verify LL^T = Sigma
#--------------

#-------------------------------------------------------------------------
```

```r
### Bootstrapping Part B – 2

## 2. For a given sample x_1,...,x_n from a mvn, estimate the mean
## vector and covariance matrix from maximum likelihood.

mvn_mle_est = function(x){
    #PURPOSE: For a sample of x1...xn ~ mvn(mu,Sigma),
    #               estimate mu and Sigma using mle estimates.

    #INPUT:     x = matrix of values.  Each row must be a sample.
    #               Each col must be an x_i ~ mvn, s.t. x is (pxn).
    #OUTPUT:    mu_hat = mle estimate of mvn mean
    #           Sig_hat = mle estimate of mvn cov matrix

    #Note: MLE estimates derived and calculated.
    #mu_hat = colMeans(x)
    #sigma_hat = cov(x)
    require(mvnmle)

    est = mlest(x)
    return(list(muhat=est$muhat,sigmahat=est$sigmahat,loglik=est$value))
}

# Generate some simulated data to work with.
library(MASS)
n = 1000
mu = c(4,7)
Sigma <- matrix(c(10,3,3,2),2,2)
x = mvrnorm(n=n,mu=mu,Sigma=Sigma)

mvn_mle_est(x)  #Output MLE estimates mu_hat, Sigma_hat.
mu              #Output true mu.
Sigma           #Output true Sigma

#-------------------------------------------------------------------------
### Bootstrapping Part B – 3

## 3. Bootstrap a given sample x_1...x_n to estimate the sampling
## distribution of the MLE.

# Generate some simulated data to work with.
library(MASS)
n = 10
mu = c(4,7)
p = length(mu)
Sigma <- matrix(c(10,3,3,2),2,2)
x = mvrnorm(n=n,mu=mu,Sigma=Sigma)

B = 10000                           #Number of bootstrap samples.
mu_boot = array(0,c(p,B))       #px1 array to hold mu vectors.
sig_boot = array(0,c(p,p,B))    #pxpxB array to hold cov matrices.

for (b in 1:B){
    #Rows of x to sample.
    rows_boot = sample(1:n,size=n,replace=T)

    #Save bootstrap sample of x.
    xb = x[rows_boot,]
```

```
    #Save bootstrap calculations for mu and Sigma.
    mu_boot[,b] = colMeans(xb)
    sig_boot[,,b] = cov(xb)
}

mu_boot_mean = apply(mu_boot,1,mean)
sig_boot_mean = apply(sig_boot,c(1,2),mean)

#Plot outputs of each parameter to observe sampling distribution.
#d=2, so two mu params, and four sigma params.
jpeg('/Users/jennstarling/UTAustin/2017S_Stats Modeling 2/Exercise-01/R Files/boot_mu.
    jpg')
par(mfrow=c(1,2))
hist(mu_boot[1,])
hist(mu_boot[2,])
dev.off()

jpeg('/Users/jennstarling/UTAustin/2017S_Stats Modeling 2/Exercise-01/R Files/boot_
    sigma.jpg')
par(mfrow=c(2,2))
hist(sig_boot[1,1,])
hist(sig_boot[1,2,])
hist(sig_boot[2,1,])
hist(sig_boot[2,2,])
dev.off()
```