Assignment 9

Jake Brawer

December 1, 2017

Problem 1

a)

First lets solve for μ_Y

$$\begin{aligned} Y &= \alpha + \beta x + \epsilon \\ \mu_Y &= E[Y] \\ &= E[\alpha + \beta x + \epsilon] \\ &= \alpha + \beta \mu_Y \end{aligned} \tag{1}$$

Using (1) we can solve for ρ

$$\begin{aligned} \rho &= \tfrac{cov(X,Y)}{\sigma_X \sigma_X} \\ &= \mathrm{E}\left[(Y - \mu_Y)(X - \mu_X)\right] \\ &= \mathrm{E}\left[(\alpha\beta X + \epsilon - \alpha + \beta\mu_X)(X - \mu_X)\right] \\ &= \mathrm{E}\left[\beta(X - \mu_X)^2 + \epsilon(X - \mu_X)\right] \\ &= \frac{\beta\sigma_X^2}{\sigma_X \sigma_Y} \\ &= \frac{\beta\sigma_X}{\sigma_Y} \end{aligned}$$

From (2) It's easy to see that

$$\beta = \rho \frac{\sigma_Y}{\sigma_X}$$

and substituting (2) in for β in (1)

$$\alpha = \mu_Y - \rho \frac{\sigma_Y}{\sigma_X} \mu_X$$

b)

$$\begin{aligned}
\sigma_Y &= E\left[(Y - \mu_Y)^2\right] \\
&= \mathrm{E}\left[(\alpha + \beta X + \epsilon - \alpha - \beta\mu_x)^2\right] \\
&= \mathrm{E}\left[(\beta(X - \mu_X) + \epsilon)^2\right] \\
&= \beta E\left[(X - \mu_X)^2\right] + E\left[\epsilon^2\right] + E\left[2\beta\epsilon(X - \mu_X)\right] \\
&= \beta^2\sigma_X^2 + \mu_\epsilon^2 + 0
\end{aligned}$$

So from here we have

$$\begin{aligned}
\sigma_\epsilon^2 &= \sigma_Y^2 - \beta^2\sigma_X^2 \\
&= \sigma_Y^2 - \rho\frac{\sigma_Y^2}{\sigma_X^2}\sigma_X^2 \\
&= (1 - \rho)^2\sigma_Y^2
\end{aligned}$$

c)

$$\begin{aligned}\tilde{\epsilon} &= \tilde{Y} - \rho\tilde{X} \\ &= \frac{\beta(X-\mu_X)+\epsilon}{\sigma_X} - \frac{\beta\sigma_X(X-\mu_X)}{\sigma_X\sigma_Y} \\ &= \frac{\epsilon}{\sigma_Y}\end{aligned}$$

Now we can show that $var(\tilde{\epsilon}) = (1-\rho)^2$

$$\begin{aligned}\sigma_{\tilde{\epsilon}} &= E\left[(\tilde{\epsilon}-\mu_\epsilon)^2\right] \\ &= \tfrac{1}{\sigma_Y} E\left[(\epsilon-\mu_\epsilon)^2\right] \\ &= \frac{\sigma_\epsilon}{\sigma_Y}\end{aligned}$$

So we have

$$\begin{aligned}\sigma^2_{\tilde{\epsilon}} &= \frac{\sigma^2_\epsilon}{\sigma^2_Y} \\ &= (1 - \rho)^2 \qquad \text{By problem b)}\end{aligned}$$

Problem 2

a)

$$\left(\frac{68-65}{3.5}\right) = .85$$

So

$$1 - \Phi(.85) = 1 - .30 = .70.$$

b)

$$\left(\frac{\hat{Y} - 65}{3.5}\right) = .3\left(\frac{\hat{X} - 70}{4.0}\right)$$

So

$$\hat{Y} = .2625\left(\hat{X} - 70\right) + 65$$

c)

$= .2625(72 - 70) + 65$

$= 65.525$

$= \mu$

$\sigma = \sqrt{1 - \rho^2}\ \sigma_Y^2$

$= \sqrt{1 - .3^2}\ 3.5^2$
$= 11.1556$

So

$$(Y|X = 72.0) \sim N(65.525, 11.1556)$$

d)

$$\left(\frac{68 - 65.525}{3.34}\right) = .74$$

Problem 3

```
library(MASS) # for truehist function
library(rjags)
salary.dat <- read.csv(
  "http://www.stat.yale.edu/~jtc5/238/data/SalariesAndGender.csv"
)
attach(salary.dat)

male <- as.numeric(gender=="m")



m3 <- "
model{
  for(i in 1:12){
    salary[i] ~ dnorm(a + b[1]*male[i] + b[2]*experience[i] + b[3]*male[i]*experience[i
  }
  a ~ dnorm(0.0, 1.0E-14)
  for(i in 1:3){b[i] ~ dnorm(0.0, 1.0E-14)}
  tau ~ dgamma(.01,.01)
}
"

jmlog <- jags.model(
  textConnection(m3),
  data=list(salary=log(salary), male=male, experience=experience)
)
jm <- jags.model(
  textConnection(m3),
  data=list(salary=salary, male=male, experience=experience)
)

update(jm, 10000)
update(jmlog, 10000)

s <- coda.samples(jm, c("a","b","tau"), 100000)
slog <- coda.samples(jmlog, c("a","b","tau"), 100000)
```

```
ss <- as.data.frame(s[[1]])
sslog <- as.data.frame(slog[[1]])
```

The likelihood that there is a positive interaction in the salary case is:

```
mean(ss$'b[3]' > 0)
```

```
0.99058
```

The likelihood that there is a positive interaction in the log(salary) case is:

```
mean(sslog$'b[3]' > 0)
```

```
0.56661
```

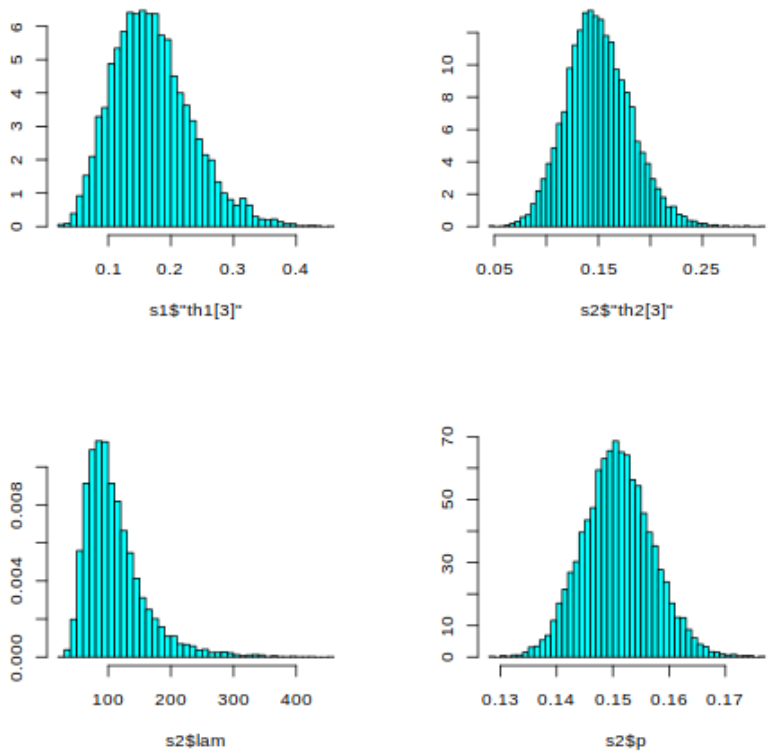
Using the log scale, it is unclear whether the interaction effect is present. Logarithmic scales are nice when you are dealing with data that spans orders of magnitude. In terms of salaries, such vasts differences in salaries are not likely to exists between employees, and so the using a log scale is thus not very useful.

Problem 4

a)

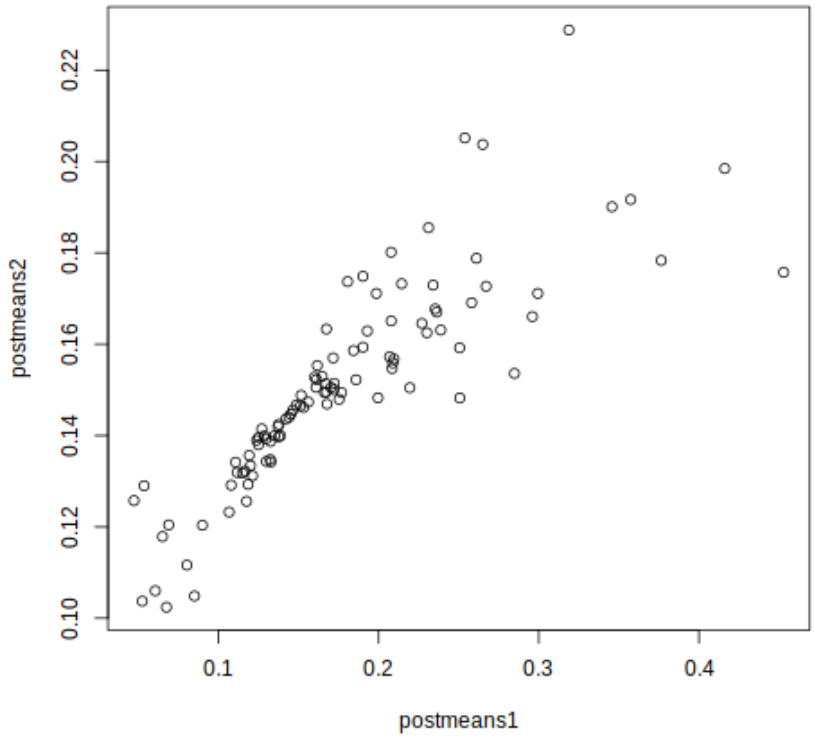
```
source("http://www.stat.yale.edu/~jtc5/238/data/martian-basketball-data.r")
m1 <- "
  model{
    for(i in 1:100){
      ks[i] ~ dbinom( th1[i], ns[i])
      th1[i] ~ dunif(0,1)
    }
}
"

m2 <- "
  model{
    for(i in 1:100){
      ks[i] ~ dbinom(th2[i], ns[i])
```

```
      th2[i] ~ dbeta(a, b)
    }
    p ~ dunif(0, 1)
    lam ~ dexp(0.0001)
    a <- lam * p
    b <- (1 - p) * lam
}
"
jm1 <- jags.model (file = textConnection ( m1 ),
                   data=list(ks=ks, ns=ns),
                   )
cs1 <- coda.samples (jm1 , c("th1"), 10000)
s1 <- as.data.frame (cs1 [[1]])

jm2 <- jags.model (file = textConnection ( m2 ),
                   data=list(ks=ks, ns=ns),
                   )
cs2 <- coda.samples (jm2 , c("th2", "p", "lam"), 10000)
s2 <- as.data.frame (cs2 [[1]])

par(mfrow = c(2,2))
truehist(s1$"th1[3]")
truehist(s2$"th2[3]")
truehist(s2$"lam")
truehist(s2$"p")
```



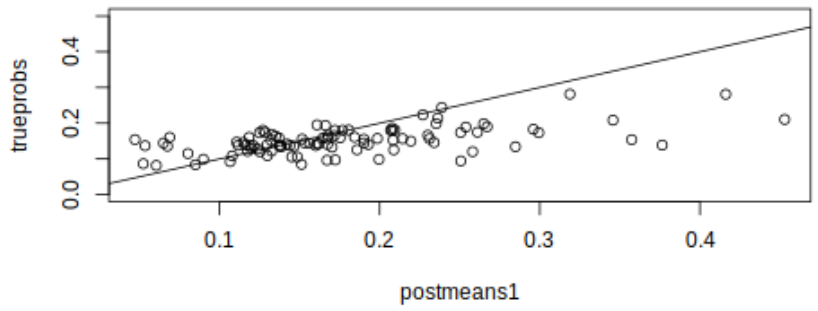
b)

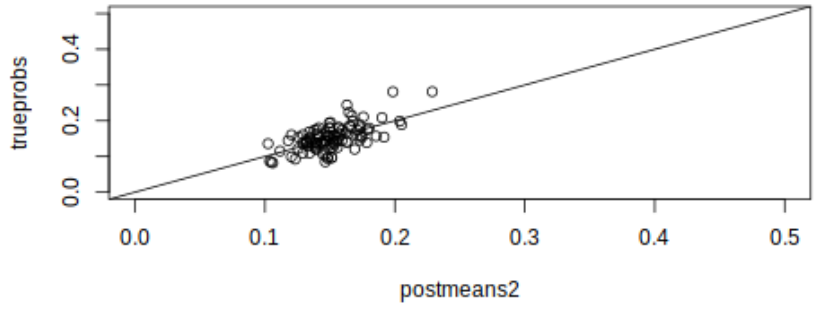
```
postmeans1 <-  colMeans(s1)
postmeans2 <-  colMeans(s2[, 3:102])

plot(postmeans1,postmeans2)
```



c)

```
length(postmeans2)
xlim <- c(0,.5)
ylim <- c(0,.5)
par(mfrow = c(2,1))
plot(postmeans1, trueprobs, col = 1, lim = xlim, ylim = ylim)
abline(coef=c(0,1))
plot(postmeans2, trueprobs, col=1, xlim = xlim, ylim = ylim)
abline(coef=c(0,1))
```





d)

```
quant <- c(0.025, 0.975)
M1L <- lapply(s1, quantile, 0.025)
M2L <- lapply(s2[3:102], quantile, 0.025)
M1U <- lapply(s1, quantile, 0.975)
M2U <- lapply(s2[3:102], quantile, 0.975)

M1L[[1]] < M1U[[1]]
nms <- c("trueprobs", "M1L", "M1U", "cover1", "M2L", "M2U", "cover2")
df <- data.frame(matrix(ncol=length(nms), nrow = 100))
colnames(df) <- nms
for(i in 1:100){
  c1 <-  trueprobs[i] >= M1L[[i]] & trueprobs[i] <= M1U[[i]]
  c2 <-  trueprobs[i] >= M2L[[i]] & trueprobs[i] <= M2U[[i]]
```

```
  r <- list(trueprobs[i],M1L[[i]], M1U[[i]], c1[[1]], M2L[[i]], M2U[[i]], c2[[1]] )
  df[i,] <- r
}
head(df)
tail(df)
```

```
  trueprobs        M1L       M1U cover1        M2L       M2U cover2
1 0.1341175 0.09808346 0.1869452   TRUE 0.10374789 0.1806429   TRUE
2 0.1732624 0.04940630 0.2321773   TRUE 0.08373924 0.1974685   TRUE
3 0.1326052 0.06724363 0.3122853   TRUE 0.09346874 0.2179882   TRUE
4 0.1443848 0.01385996 0.1535057   TRUE 0.06411825 0.1749340   TRUE
5 0.1585323 0.08861564 0.2551024   TRUE 0.10135107 0.2103108   TRUE
6 0.1808951 0.07434845 0.3932984   TRUE 0.09771855 0.2323512   TRUE
     trueprobs        M1L       M1U cover1        M2L       M2U cover2
95  0.15666284 0.12647688 0.3153460   TRUE 0.12019647 0.2376223   TRUE
96  0.21049344 0.19377363 0.7359731   TRUE 0.10960895 0.2644518   TRUE
97  0.19851406 0.18027855 0.3578279   TRUE 0.14806369 0.2724698   TRUE
98  0.14461626 0.04449855 0.3134686   TRUE 0.08547946 0.2139191   TRUE
99  0.17478313 0.14492910 0.4076666   TRUE 0.12030734 0.2578638   TRUE
100 0.09875627 0.02843607 0.4858974   TRUE 0.08558720 0.2292309   TRUE
```

e)

```
length(df$cover1[df$cover1==TRUE]) /100
```

```
0.95
```

```
length(df$cover2[df$cover2==TRUE]) /100
```

```
0.95
```

the predicted value is within the interval \$95\$% of the time for both models, so its hard to judge on those terms alone. Lets look at the average size of each interval

```
mu1 <- mean(df$M1U - df$M1L)
sd1 <- sd(df$M1U - df$M1L)
sprintf("Mean interval length for M1: %s, SD: %s", mu1, sd1)
```

```
Mean interval length for M1: 0.222794270035701, SD: 0.123643398032502
```

```
mu2 <- mean(df$M2U - df$M2L)
sd2 <- sd(df$M2U - df$M2L)
sprintf("Mean interval length for M2: %s, SD: %s", mu2, sd2)
```

```
Mean interval length for M2: 0.113894949022019, SD: 0.023634332314039
```

Clearly model the set of intervals from Model 2 are preferred. While M2 is no more accurate than M1, the intervals more precisely hone in on the predicted value.

f)

First we will store the best players for each model.

```
I1 <- rep(0, dim(s1)[1])
I2 <- rep(0, dim(s1)[1])
ths2 <- subset(s2,select = -c(1,2))
# Save the best player at each iteration for each model.
for(i in 1:dim(s1)[1]){
  I1[i] <- which(s1[i, ] == max(s1[i, ]))
  I2[i] <- which(ths2[i, ] == max(ths2[i, ]))
}
```

The probability that 19 is the best player according to model 1 is

```
length(I1[I1== 19])/length(I1)
```

```
0.0029
```

The probability that 19 is the best player according to model 2 is

```
length(I2[I2== 19])/length(I2)
```

```
0.2631
```

Clearly Model 2 is the better model.