

Writing Project

Meaghan Winder

1/21/2020

Read in the data

```
library(readxl)
USBR.login <- read_excel("USBR_2018_data_sample.xlsx", sheet = 1)
USBR.sampleprep <- read_excel("USBR_2018_data_sample.xlsx", sheet = 2)
USBR.microscopy <- read_excel("USBR_2018_data_sample.xlsx", sheet = 3)
USBR.PCR <- read_excel("USBR_2018_data_sample.xlsx", sheet = 4)

unique(USBR.sampleprep$`Water Name`) # number of lakes (sites) in the 2018 sample data

## [1] "Grant Inlet (EMWD)"          "Irrp Pond (EMWD)"
## [3] "Lake Mead (Hoover Dam)"      "Apache Lake (Horse Mesa Dam)"
## [5] "Bartlett Lake"              "Canyon Lake (Mormon Flat Dam)"
## [7] "Roosevelt Lake"            "Saguaro Lake (Stewart Mountain Dam)"
## [9] "Granite Reef Diversion Dam"
```

Occupancy Models

$y_{i,j} = \{0,1\}$ is the binary response for the j^{th} sample within site i

If the site is occupied: $y_{i,j} \sim \text{Bernoulli}(p)$ $y_i \sim \text{Binomial}(J, p)$ where y_i is the total number of detections from the J samples in the i^{th} site

If the site is unoccupied: $y_{i,j} = 0$ with probability 1, since we assume there are no false detections

Simulate a data set

M sites each sampled J times constant p and ψ

```
set.seed(1202020)
p <- 0.6 # constant detection probability... could change to depend on covariates
psi <- 0.6 # constant occupancy probability... could change to depend on covariates
M <- 10 # number of sites
J <- 8 # constant number of samples per site... does not need to be constant

z <- rbinom(M, 1, psi) # site occupancy

y1 <- matrix(NA, nrow = M*J, ncol = 4)
y1[, 1] <- rep(1:M, each = J) # column indicating site
```

```

y1[, 2] <- rep(1:J, M) # column indicating sample within site
y1[, 3] <- rep(z, each = J) # column indicating true site occupancy

# creates a column of 1's and 0's indicating whether the species was detected
for(i in 1:(M*J)){
  y1[i, 4] <- rbinom(1, 1, p*y1[i, 3]) # if the site is not occupied the species cannot be detected
}

colnames(y1) <- c("Site", "Sample", "True Occupancy", "Detected")

library(car)

```

Loading required package: carData

```
some(y1) # view 10 sample rows of the simulated data
```

```
##      Site Sample True Occupancy Detected
## [1,]    2      3              0         0
## [2,]    2      8              0         0
## [3,]    4      1              1         0
## [4,]    5      3              1         0
## [5,]    7      1              0         0
## [6,]    7      4              0         0
## [7,]    8      5              0         0
## [8,]    8      7              0         0
## [9,]   10      3              0         0
## [10,]  10      4              0         0
```

or

```
y2 <- rbinom(M, J, p*z) # total number of detections for the J samples within each of the M sites, y_i
```

Using $p \sim \text{Beta}(1,1)$ or $\text{Unif}(0,1)$ (that's what the Royle book says but I think it's the Beta one, but I think you need Beta dist. to find the posterior distributions described) and $\psi \sim \text{Beta}(1,1)$, and when $y_i > 0$ we know that $z_i = 1$, but when $y_i = 0$ we need to take a draw from the conditional distribution of $z_i | y_i = 0 \sim \text{Bernoulli}(\frac{\psi(1-p)^J}{\psi(1-p)^J + (1-\psi)})$, then conditional posterior distribution of p is $\text{Beta}(a_p, b_p)$ where $a_p = 1 + \sum_{i=1}^M y_i z_i$ and $b_p = 1 + J \sum_{i=1}^M z_i - \sum_{i=1}^M y_i z_i$, the conditional posterior distribution of ψ is $\text{Beta}(a_\psi, b_\psi)$ where $a_\psi = 1 + \sum_{i=1}^M z_i$ and $b_\psi = 1 + M - \sum_{i=1}^M z_i$.

```

y <- y2
M <- length(y)
J <- 8 # from above

nsim <- 10000

psi <- rep(NA, nsim)
p <- rep(NA, nsim)

psi[1] <- 0.5 # starting value for psi
p[1] <- 0.5 # starting value for p

for(k in 2:nsim){
  z <- rep(NA, M)
  for(i in 1:M){
    if(y[i] > 0){

```

```

    z[i] <- 1
  } else{
    prob <- (psi[k-1]*(1-p[k-1])^J)/(psi[k-1]*(1-p[k-1])^J + (1-psi[k-1]))
    z[i] <- rbinom(1, 1, prob)
  }
}
a_p <- 1 + sum(y*z)
b_p <- 1 + J*sum(z) - sum(y*z)
p[k] <- rbeta(1, a_p, b_p)

a_psi <- 1 + sum(z)
b_psi <- 1 + M - sum(z)
psi[k] <- rbeta(1, a_psi, b_psi)
}

quantile(psi, c(0.025, 0.975))

```

```

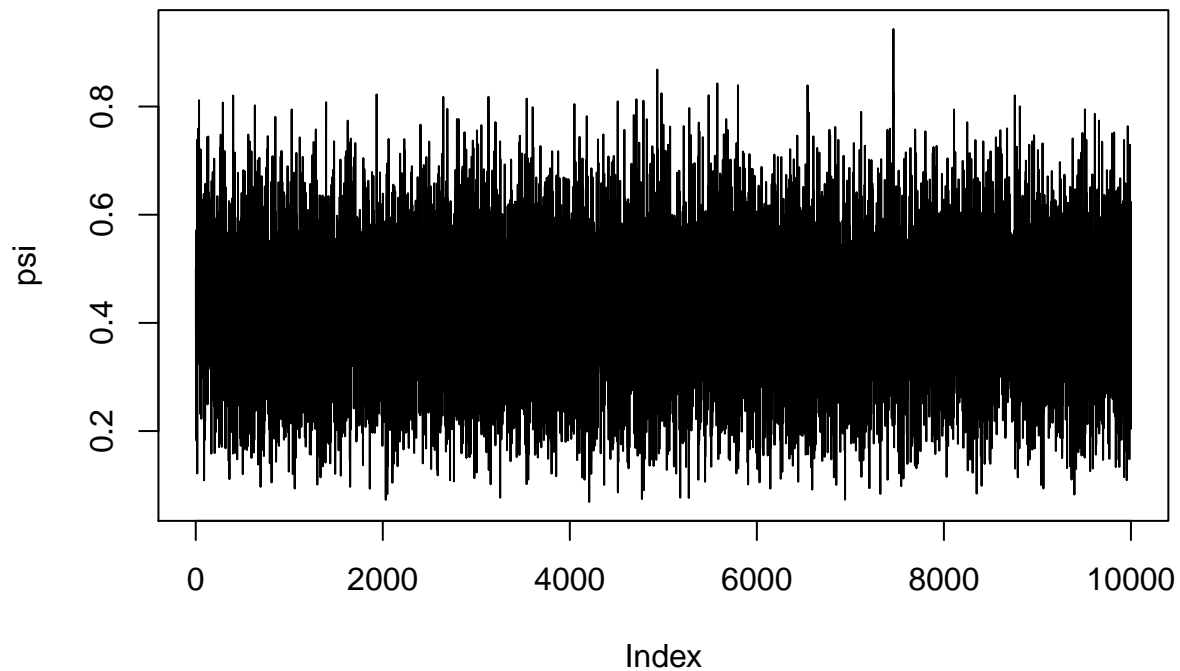
##      2.5%      97.5%
## 0.1695138 0.6928794

```

```
mean(psi)
```

```
## [1] 0.4179555
```

```
plot(psi, type = 'l')
```



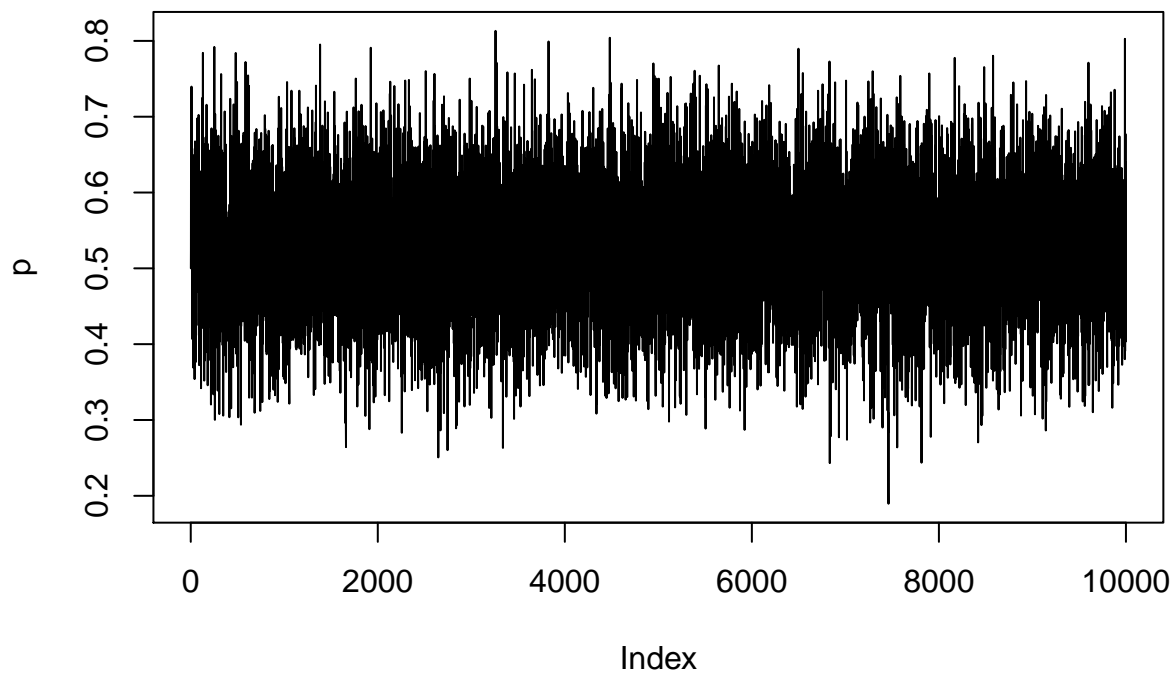
```
quantile(p, c(0.025, 0.975))
```

```
##      2.5%      97.5%  
## 0.3613128 0.6905305
```

```
mean(p)
```

```
## [1] 0.5274889
```

```
plot(p, type = 'l')
```



Simulate another data set

M sites each sampled j_i times constant p and ψ

```
set.seed(1222020)  
p <- 0.6 # constant detection probability... could change to depend on covariates  
psi <- 0.6 # constant occupancy probability... could change to depend on covariates  
M <- 10 # number of sites  
J <- sample(1:10, M, replace = T) # number of times each of the sites were sampled  
  
z <- rbinom(M, 1, psi) # site occupancy  
  
y <- rep(NA, M)  
  
for(i in 1:M){
```

```
y[i] <- rbinom(1, J[i], p*z[i]) # total number of detections for the J samples within each of the M s
}
```

Using $p \sim \text{Beta}(1,1)$ or $\text{Unif}(0,1)$ (that's what the Royle book says but I think it's the Beta one, but I think you need Beta dist. to find the posterior distributions described) and $\psi \sim \text{Beta}(1,1)$, and when $y_i > 0$ we know that $z_i = 1$, but when $y_i = 0$ we need to take a draw from the conditional distribution of $z_i|y_i = 0 \sim \text{Bernoulli}(\frac{\psi(1-p)^{j_i}}{\psi(1-p)^{j_i} + (1-\psi)})$, then conditional posterior distribution of p is $\text{Beta}(a_p, b_p)$ where $a_p = 1 + \sum_{i=1}^M y_i z_i$ and $b_p = 1 + \sum_{i=1}^M j_i z_i - \sum_{i=1}^M y_i z_i$, the conditional posterior distribution of ψ is $\text{Beta}(a_\psi, b_\psi)$ where $a_\psi = 1 + \sum_{i=1}^M z_i$ and $b_\psi = 1 + M - \sum_{i=1}^M z_i$.

```
y <- y
M <- length(y)
J <- J

nsim <- 10000

psi <- rep(NA, nsim)
p <- rep(NA, nsim)

psi[1] <- 0.5 # starting value for psi
p[1] <- 0.5 # starting value for p

for(k in 2:nsim){
  z <- rep(NA, M)
  for(i in 1:M){
    if(y[i] > 0){
      z[i] <- 1
    } else{
      prob <- (psi[k-1]*(1-p[k-1])^J[i])/(psi[k-1]*(1-p[k-1])^J[i] + (1-psi[k-1]))
      z[i] <- rbinom(1, 1, prob)
    }
  }
  a_p <- 1 + sum(y*z)
  b_p <- 1 + sum(J*z) - sum(y*z)
  p[k] <- rbeta(1, a_p, b_p)

  a_psi <- 1 + sum(z)
  b_psi <- 1 + M - sum(z)
  psi[k] <- rbeta(1, a_psi, b_psi)
}

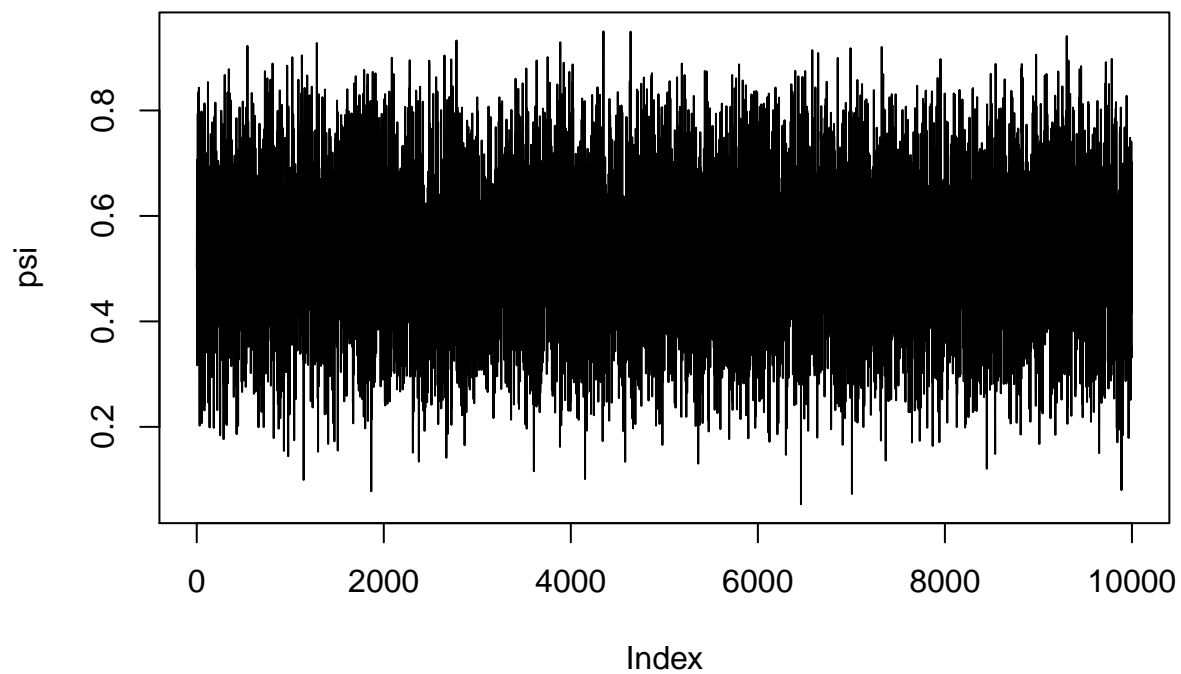
quantile(psi, c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 0.2503961 0.8092382
```

```
mean(psi)
```

```
## [1] 0.5322064
```

```
plot(psi, type = 'l')
```



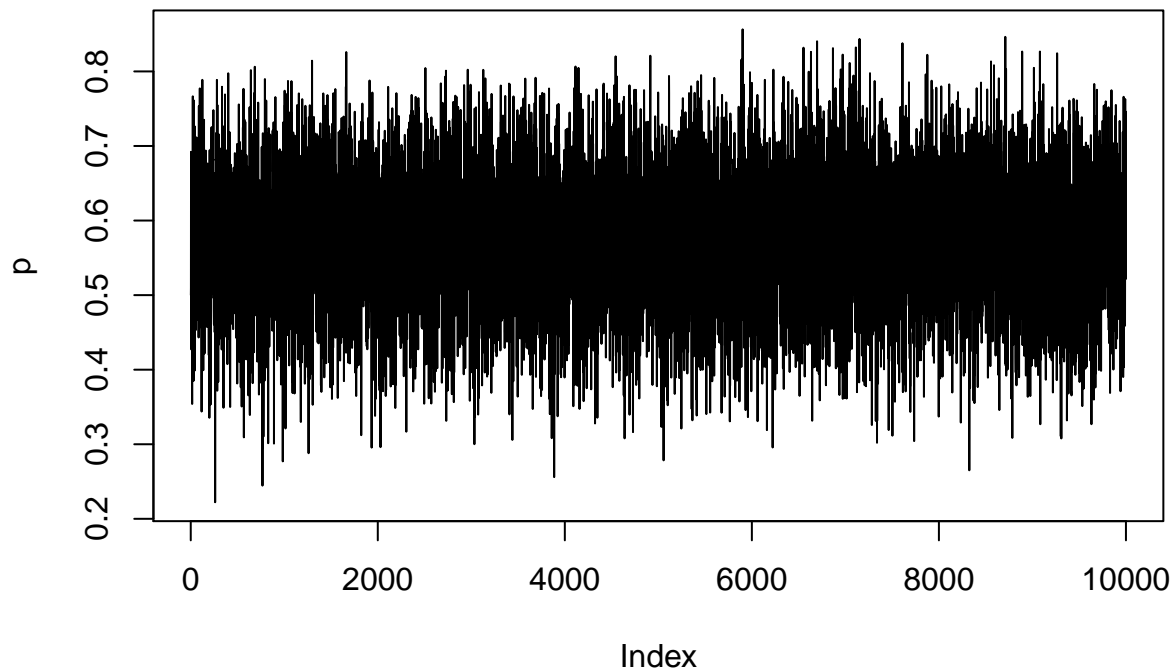
```
quantile(p, c(0.025, 0.975))
```

```
##      2.5%      97.5%  
## 0.3907857 0.7489625
```

```
mean(p)
```

```
## [1] 0.5740904
```

```
plot(p, type = 'l')
```



Simulate another data set

M sites each sampled j_i times p and ψ both depend on site level covariate x

```
set.seed(1222020)
M <- 10 # number of sites
x <- runif(10, 0, 10)
p <- exp(2 - 0.5*x)/(1 + exp(2 - 0.5*x)) # detection probability depends on site covariate x
psi <- exp(0.5 + 0.2*x)/(1 + exp(0.5 + 0.2*x)) # occupancy probability depends on site covariate x
J <- sample(1:10, M, replace = T) # number of times each of the sites were sampled

z <- rep(NA, M)
y <- rep(NA, M)

for(i in 1:M){
  z[i] <- rbinom(1, 1, psi[i]) #site occupancy
  y[i] <- rbinom(1, J[i], p[i]*z[i]) # total number of detections for the J samples within each of the sites
}
```

$\text{logit}(p_i) = \beta_1 + \beta_2 * x_i$ and $\text{logit}(\psi_i) = \beta_3 + \beta_4 * x_i$ so now we need to use a Metropolis algorithm

```
library(mnormt) # needed if you use `rmnorm`
set.seed(01282020)

X <- matrix(c(rep(1, length(x)), x), ncol = 2)
```

```
par <- dim(X)[2]
```

```
num_mcmc <- 10000
```

```
step_size <- rep(0.01, par)
```

```
# accept_ratio_p <- matrix(0, nrow = num_mcmc, ncol = par)
```

```
# accept_ratio_psi <- matrix(0, nrow = num_mcmc, ncol = par)
```

```
# p_beta_mcmc <- matrix(0, nrow = num_mcmc, ncol = par)
```

```
# psi_beta_mcmc <- matrix(0, nrow = num_mcmc, ncol = par)
```

```
#
```

```
# p_beta_prior_var <- diag(par)*10000
```

```
# psi_beta_prior_var <- diag(par)*10000
```

```
# p_beta_current <- p_beta_mcmc[1, ]
```

```
# psi_beta_current <- psi_beta_mcmc[1, ]
```

```
#
```

```
# for(i in 2:num_mcmc){
```

```
#   for(j in 1:par){
```

```
#     p_beta_star <- p_beta_current
```

```
#     p_beta_star[j] <- p_beta_star[j] + rnorm(1, 0, step_size[j])
```

```
#
```

```
#     psi_beta_star <- psi_beta_current
```

```
#     psi_beta_star[j] <- psi_beta_star[j] + rnorm(0, 1, step_size[j])
```

```
#
```

```
#     psi_current <- exp(X %*% psi_beta_current)/(1 + exp(X %*% psi_beta_current))
```

```
#     psi_star <- exp(X %*% psi_star)/(1 + exp(X %*% psi_star))
```

```
#
```

```
#     log_pi_psi_current <-
```

```
#
```

```
#     p_current <- exp(X %*% p_beta_current)/(1 + exp(X %*% p_beta_current))
```

```
#     p_star <- exp(X %*% p_beta_star)/(1 + exp(X %*% p_beta_star))
```

```
#
```

```
#     log_pi_p_current <-
```

```
#
```

```
#
```

```
#   }
```

```
# }
```

```
# I am confused now because I got near the end and then realized that the betas associated with p and p
```

```
# I need help now
```