

# Problem Set 1 Stout

Brennan Stout

2025-09-02

## 1. Independence

Rewrite Bayes' Law when the two events are independent. How do you interpret this?

---

Bayes' Law

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- $P(A|B) = P(A)$
- $P(B|A) = P(B)$

$$P(A|B) = \frac{P(B) \cdot P(A)}{P(B)} = P(A)$$

The occurrence of A does not impact the occurrence of B as both are independent. Given this the probability of A given B is equal to the probability of A alone.

## 2. Gibbs

Using R, re-run the Gibbs sampling function given in class for the bivariate exponential. What effect do you see in varying the  $B$  parameter? What is the effect of producing 200 sampled values instead of 50000?

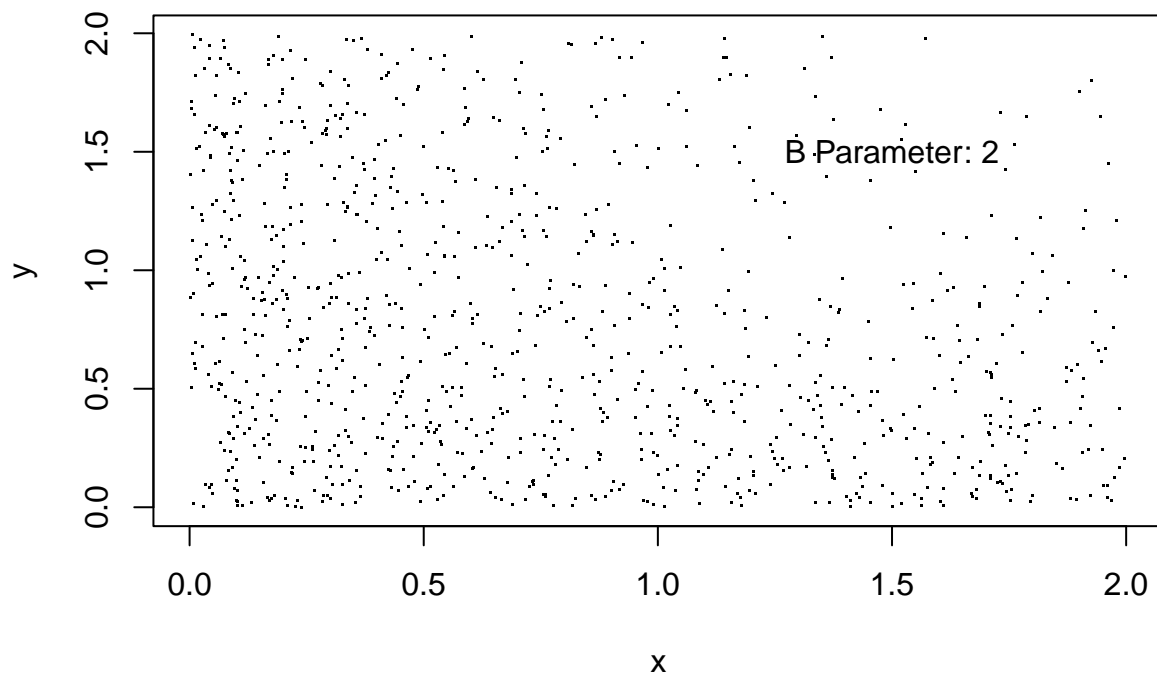
---

The  $B$  parameter controls the extent of the x and y axis as it sets the upper bound of the joint density. A larger  $B$  will extend the range of the density while a lower values will constrain it.

```
set.seed(1)
B<-2; k<-15; m<-1000; x<-NULL; y<-NULL # Set variables
while(length(x)<m) # Gibbs Sampler
{
  x.val<-c(runif(1,0,B),rep((B+1),length=k))
  y.val<-c(runif(1,0,B),rep((B+1),length=k))
  for(j in 2:(k+1))
  {
    while(x.val[j]>B)x.val[j]<-rexp(1,y.val[j-1])
    while(y.val[j]>B)y.val[j]<-rexp(1,x.val[j])
  }
  x<-c(x,x.val[(k+1)])
  y<-c(y,y.val[(k+1)])
}

plot(x,y, pch=".", main=paste0("Path of the Gibbs Sampler, samples: ", m))
text(B/2 + B/4, B/2 + B/4, paste0("B Parameter: ", B))
```

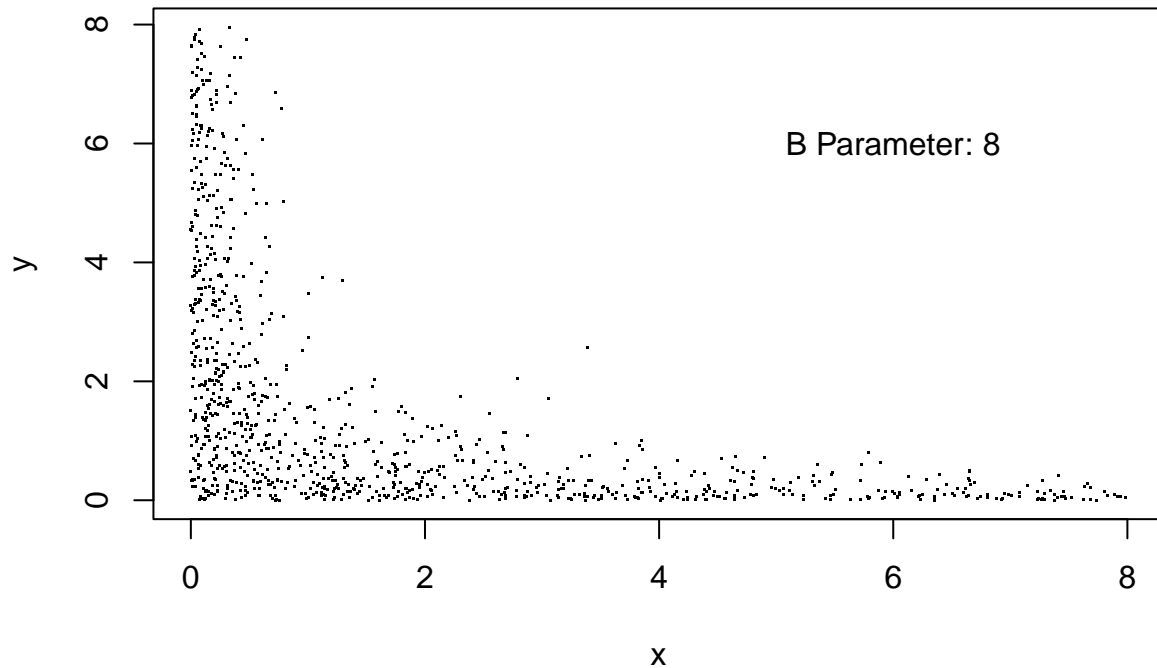
## Path of the Gibbs Sampler, samples: 1000



```
set.seed(1)
B<-8; k<-15; m<-1000; x<-NULL; y<-NULL # Set variables
while(length(x)<m) # Gibbs Sampler
{
  x.val<-c(runif(1,0,B),rep((B+1),length=k))
  y.val<-c(runif(1,0,B),rep((B+1),length=k))
  for(j in 2:(k+1))
  {
    while(x.val[j]>B)x.val[j]<-rexp(1,y.val[j-1])
    while(y.val[j]>B)y.val[j]<-rexp(1,x.val[j])
  }
  x<-c(x,x.val[(k+1)])
  y<-c(y,y.val[(k+1)])
}

plot(x,y, pch=".", main=paste0("Path of the Gibbs Sampler, samples: ", m))
text(B/2 + B/4, B/2 + B/4, paste0("B Parameter: ", B))
```

## Path of the Gibbs Sampler, samples: 1000

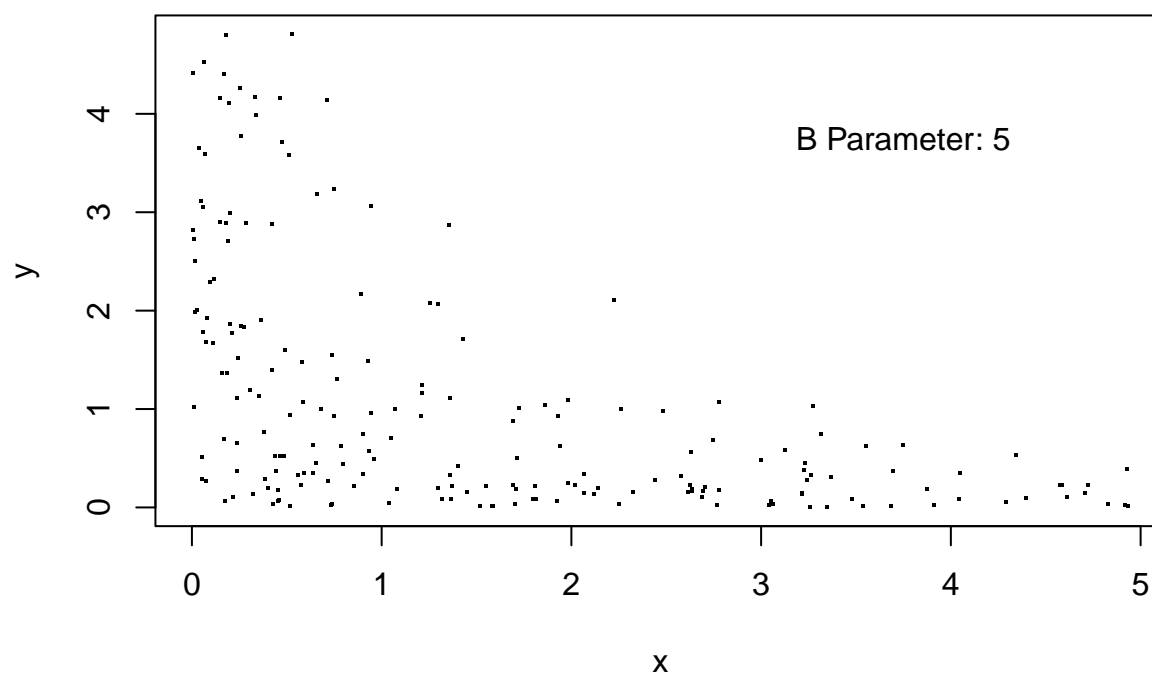


When modifying the  $m$  value this will set the number of samples used for the function. Unsurprisingly, more samples will equate to a clearer picture, but given the nature of the Gibbs algorithm moves will have more error.

```
set.seed(1)
B<-5; k<-15; m<-200; x<-NULL; y<-NULL # Set Variables
while(length(x)<m) # Gibbs Sampler
{
  x.val<-c(runif(1,0,B),rep((B+1),length=k))
  y.val<-c(runif(1,0,B),rep((B+1),length=k))
  for(j in 2:(k+1))
  {
    while(x.val[j]>B)x.val[j]<-rexp(1,y.val[j-1])
    while(y.val[j]>B)y.val[j]<-rexp(1,x.val[j])
  }
  x<-c(x,x.val[(k+1)])
  y<-c(y,y.val[(k+1)])
}

plot(x,y, pch=".", main=paste0("Path of the Gibbs Sampler, samples: ", m), cex = 2)
text(B/2 + B/4, B/2 + B/4, paste0("B Parameter: ", B))
```

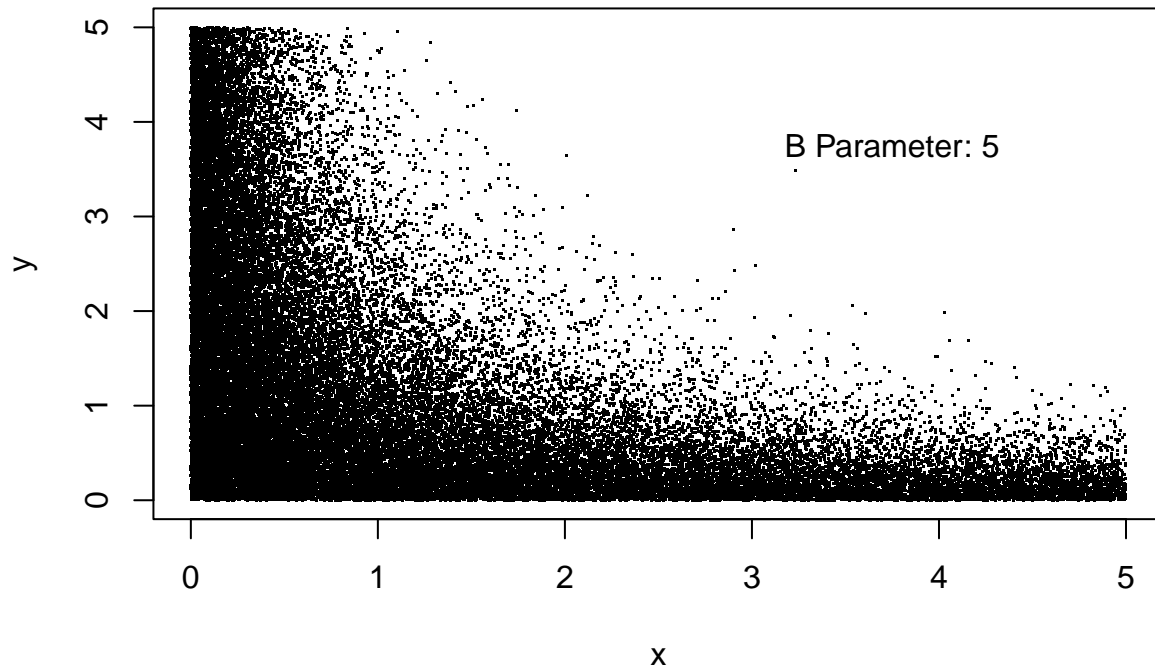
## Path of the Gibbs Sampler, samples: 200



```
set.seed(1)
B<-5; k<-15; m<-50000; x<-NULL; y<-NULL # Set variables
while(length(x)<m) # Gibbs Sampler
{
  x.val<-c(runif(1,0,B),rep((B+1),length=k))
  y.val<-c(runif(1,0,B),rep((B+1),length=k))
  for(j in 2:(k+1))
  {
    while(x.val[j]>B)x.val[j]<-rexp(1,y.val[j-1])
    while(y.val[j]>B)y.val[j]<-rexp(1,x.val[j])
  }
  x<-c(x,x.val[(k+1)])
  y<-c(y,y.val[(k+1)])
}

plot(x,y, pch=".", main=paste0("Path of the Gibbs Sampler, samples: ", m))
text(B/2 + B/4, B/2 + B/4, paste0("B Parameter: ", B))
```

## Path of the Gibbs Sampler, samples: 50000



### 3. M-H

Implement the Gibbs and Metropolis sampling algorithms for the bivariate exponential sampling problem as noted above. Plot the paths of the samplers in X-Y space and compare the results. What is the difference in the sampling paths? (consult Gill pages 351–60. for another example). Also, look at the R functions `points` and `segments` for drawing the paths. You will probably only want to let this run for say 200 iterations or less.

From the results, it is apparent that the Gibbs samplers will spread out its steps more. Since this sampler must move with each step the generated points, while biased towards the left they are spread across the entire test space. For Metropolis, since this sampler has the option to not move with each step the resulting points will remain more consistent in their distribution. This is reflected in the results, as the M-H points are much more clustered.

```
set.seed(1)

# Target density function for bivariate exponential
target_density <- function(x, y) {
  if (x < 0 || y < 0) return(0)
  return(exp(-x - y - x * y))
}

B <- 5 # Set B value for Gibbs
k <- 15
num.iterations <- 200 # Number of samples
x <- NULL; y <- NULL
omega <- 3/4 # Set omega for MH
```

```

# Gibbs Sampler
while(length(x) < num.iterations)
{
  x.val<-c(runif(1,0,B),rep((B+1),length=k))
  y.val<-c(runif(1,0,B),rep((B+1),length=k))
  for(j in 2:(k+1))
  {
    while(x.val[j]>B)x.val[j]<-rexp(1,y.val[j-1])
    while(y.val[j]>B)y.val[j]<-rexp(1,x.val[j])
  }
  x<-c(x,x.val[(k+1)])
  y<-c(y,y.val[(k+1)])
}

# Initialize chain
x.y <- matrix(c(1, 1), 1, 2, dimnames = list("1", c("x", "y")))
# Metropolis loop
for (t in 2:num.iterations) {
  candidate <- x.y[t - 1, ] + omega * rnorm(2, 0, 0.25)

  # Reject if candidate is out of domain
  if (candidate[1] < 0 || candidate[2] < 0) {
    x.y <- rbind(x.y, x.y[t - 1, ])
  } else {
    accept.prob <- min(1, target_density(candidate[1], candidate[2]) /
                      target_density(x.y[t - 1, 1], x.y[t - 1, 2]))

    if (accept.prob > runif(1)) {
      x.y <- rbind(x.y, candidate)
    } else {
      x.y <- rbind(x.y, x.y[t - 1, ])
    }
  }

  dimnames(x.y)[[1]][t] <- as.character(t)
}
###
# Plot both paths
plot(x, y, pch = "+", col = "blue", cex = 0.8, main = "M-H and Gibbs Sampling Paths")
for(i in 2:(length(x)))
{
  segments(x[i], y[i-1], x[i+1], y[i-1], cex = 0.1, col = "lightblue")
  segments(x[i+1], y[i], x[i+1], y[i-1], cex = 0.1, col = "lightblue")
}
points(x.y, pch = 18, cex = 0.8, col = "red")
legend("topright",
      legend = c("M-H", "Gibbs", "Gibbs Path"),
      pch = c(18, 3, NA),
      lty = c(NA, NA, 1),
      col = c("red", "blue", "lightblue"),
      cex = 0.8
)

```

## M-H and Gibbs Sampling Paths

