Generate Data:

```
set.seed(1)
n_vec <- c(100, 200)
p_vec <- c(200, 500)
snr_vec <- c(2, 4)
data_list <- vector("list", 8)
for (i in 0:1) {
    for (j in 0:1) {
        for (k in 0:1) {
            n <- n_vec[i+1]
            p <- p_vec[j+1]
            snr <- snr_vec[k+1]
            data <- get_data(n, p, snr)
            data_list[[i * 2^2 + j * 2 + k + 1]] <- list("n" = n, "p" = p, "snr" = snr, "data" = data)
        }
    }
}
```

Spike and Slab:

```
set.seed(12)

times <- rep(0, 8)
supp_sizes <- rep(0, 8)
zero_ones <- rep(0, 8)

for (i in 1:8) {
    print(i)
    data_i <- data_list[[i]]$data

    gam0 <- data_i$b0
    start_time <- Sys.time()
    gam_hat = spike_n_slab(data_i)
    end_time <- Sys.time()

    times[i] <- end_time - start_time
    zero_ones[i] <- zero_one(gam0, gam_hat)
    supp_sizes[i] <- supp_size(gam_hat)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

```
res_mat <- matrix(0, 8, 6)

for (i in 1:8) {
```

```r
    res_mat[i, 1] <- data_list[[i]]$n
    res_mat[i, 2] <- data_list[[i]]$p
    res_mat[i, 3] <- data_list[[i]]$snr
}

res_mat[ , 4] <- times
res_mat[ , 5] <- supp_sizes
res_mat[ , 6] <- zero_ones

df <- data.frame(res_mat)
names(df) <- c("n", "p", "snr", "times", "supp_sizes", "zero_ones")

df
```

```
##     n   p snr     times supp_sizes zero_ones
## 1 100 200   2  1.421297         56        46
## 2 100 200   4  1.534355         52        42
## 3 100 500   2  2.856052         67        57
## 4 100 500   4  2.262488         69        59
## 5 200 200   2 15.334744        100        90
## 6 200 200   4 15.319938        100        90
## 7 200 500   2 14.803172        100        90
## 8 200 500   4 17.965752        100        90
```

Variational Bayes:

```r
set.seed(12)

times <- rep(0, 8)
supp_sizes <- rep(0, 8)
zero_ones <- rep(0, 8)

for (i in 1:8) {
    print(i)
    data_i <- data_list[[i]]$data
    xi <- data_i$x
    yi <- data_i$y
    pi <- ncol(xi)
    gam0 <- data_i$b0

    start_time <- Sys.time()
    ind_hat = varbvs(xi, NULL, yi, verbose = F)$pip
    end_time <- Sys.time()

    gam_hat <- rep(0, pi)
    gam_hat[ind_hat] <- 1
    times[i] <- end_time - start_time
    zero_ones[i] <- zero_one(gam0, gam_hat)
    supp_sizes[i] <- supp_size(gam_hat)
}
```

```
## [1] 1
## [1] 2
```

```
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

```
res_mat <- matrix(0, 8, 6)

for (i in 1:8) {
    res_mat[i, 1] <- data_list[[i]]$n
    res_mat[i, 2] <- data_list[[i]]$p
    res_mat[i, 3] <- data_list[[i]]$snr
}

res_mat[ , 4] <- times
res_mat[ , 5] <- supp_sizes
res_mat[ , 6] <- zero_ones

df <- data.frame(res_mat)
names(df) <- c("n", "p", "snr", "times", "supp_sizes", "zero_ones")

df
```

```
##     n   p snr     times supp_sizes zero_ones
## 1 100 200   2 1.6553028          0        10
## 2 100 200   4 2.1520250          0        10
## 3 100 500   2 1.2199373          1         9
## 4 100 500   4 2.4449253          0        10
## 5 200 200   2 0.6615543          1         9
## 6 200 200   4 0.6611300          1         9
## 7 200 500   2 1.6544442          1         9
## 8 200 500   4 2.2431278          0        10
```

Horseshoe

```
# (wrapper courtesy of Felix)
set.seed(12)

times <- rep(0, 8)
supp_sizes <- rep(0, 8)
zero_ones <- rep(0, 8)

for (i in 1:8) {
    print(i)
    data_i <- data_list[[i]]$data
    xi <- data_i$x
    yi <- data_i$y
    pi <- ncol(xi)
    gam0 <- data_i$b0

    InferenceResultList <- HorseshoeMCMC(xi, yi)
    gam_hat <- PostHorseshoeMCMC(InferenceResultList)
```

```
    times[i] <- InferenceResultList$CPUTime
    zero_ones[i] <- zero_one(gam0, gam_hat)
    supp_sizes[i] <- supp_size(gam_hat)
}
```

```
## [1] 1
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 2
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 3
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 4
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 5
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 6
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
```

```
## [1] 8
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
```

```
res_mat <- matrix(0, 8, 6)

for (i in 1:8) {
    res_mat[i, 1] <- data_list[[i]]$n
    res_mat[i, 2] <- data_list[[i]]$p
    res_mat[i, 3] <- data_list[[i]]$snr
}

res_mat[ , 4] <- times
res_mat[ , 5] <- supp_sizes
res_mat[ , 6] <- zero_ones

df <- data.frame(res_mat)
names(df) <- c("n", "p", "snr", "times", "supp_sizes", "zero_ones")

df
```

```
##     n   p snr     times supp_sizes zero_ones
## 1 100 200   2  7.275271         30        36
## 2 100 200   4  7.230900         39        45
## 3 100 500   2 14.226303        108        98
## 4 100 500   4 14.109857        140       134
## 5 200 200   2 32.681750         10         0
## 6 200 200   4 31.859200         10         0
## 7 200 500   2 43.262279        109        99
## 8 200 500   4 42.667546         93        83
```