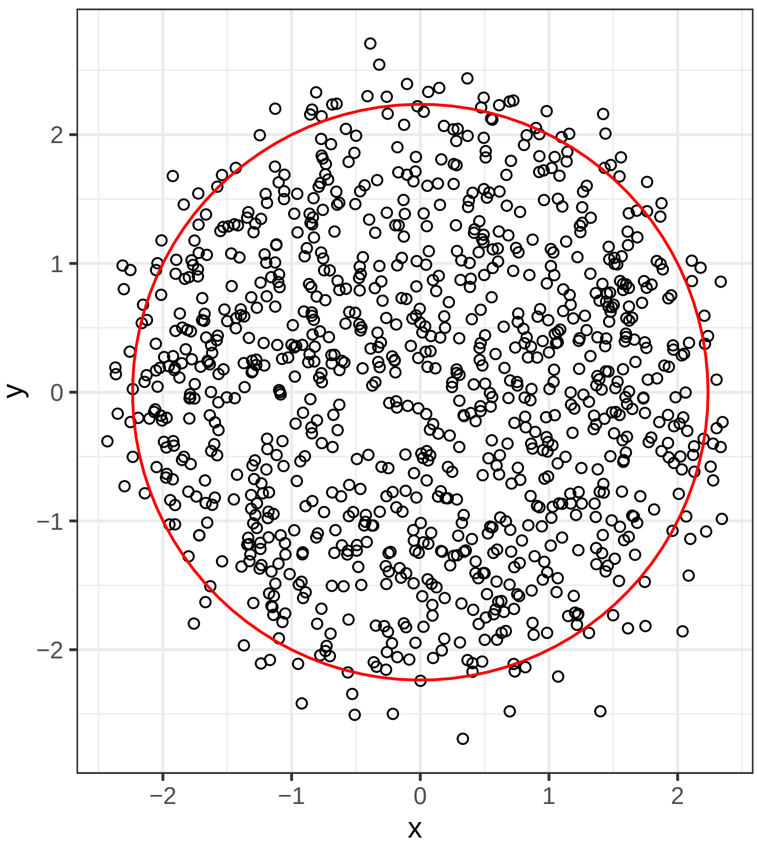
STA 602 Lab 7

Ryan Tang

01 November, 2022

Exercise 1

Here we sample from the derived full conditional equations.



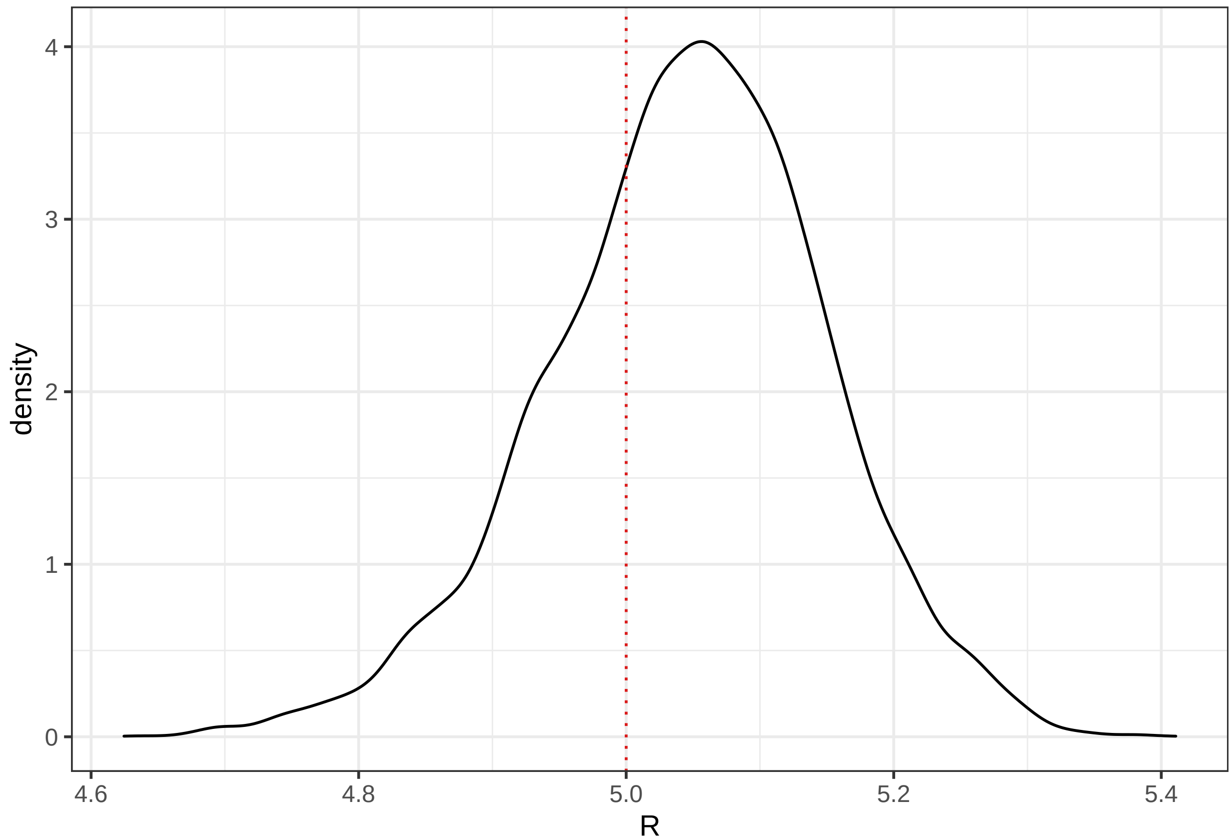

```
set.seed(23498324)

# hyper-parameters
m <- 3
k <- 1
alpha <- 5/2
beta <- 5/2

# generate random sample from pareto(m,k)
rpareto <- function(m, k, trunc = NULL){
  p <- m*(1 - runif(1))^(-1/k)
```

```r
  if(!is.null(trunc)){
    while(p > trunc){
      p <- m*(1 - runif(1))^(-1/k)
    }
  }
  return(p)
}

#
uni_pareto_gibbs <- function(S, r, m, k, alpha, beta, burn_in = min(1000, S / 2), thin = 5){
  # Reparametrize X matrix to squared radius values
  Rsq <- r  # squared distance
  n <- length(Rsq)  # sample size
  R2 <- rep(1, S)  # to save the S gibbs samples for R^2, also intialized it
  U <- matrix(0, nrow = S, ncol = n)  # to save the S gibbs samples for u_1, ..., u_n
  U[1, ] <- runif(n, 0, R2)  # initialize the first sample for u_1, ..., u_n
  sigma <- rep(1, S)  # to save the S gibbs samples for sigma (not sigma2!)

  # initialize
  U_curr <- U[1, ]  # save the updated sample for u_1, ..., u_n
  R2_curr <- R2[1]  # save the updated sample for R^2
  sigma_curr <- sigma[1]  # save the updated sample for sigma (not sigma2!)
  for(s in 1:S){
    # Sample from full conditional of the inner radius
    R2_curr <- rpareto(max(c(U_curr, m)), k + n)
    R2[s] <- R2_curr
    # Sample from full conditional of U values
    U_curr <- truncnorm::rtruncnorm(n, a = 0, b = R2_curr, mean = Rsq, sd = sigma_curr)
    U[s, ] <- U_curr
    # Sample from full conditional of sigma
    sigma_curr <-  sqrt(1 / rgamma(1, shape=n/2+alpha, rate=beta + 0.5*sum((r-U_curr)^2)))
    sigma[s] <- sigma_curr
  }
  return(list(R = R2[seq(burn_in, S, by = thin)],
              U = U[seq(burn_in, S, by = thin), ],
              sigma = sigma[seq(burn_in, S, by = thin)]))
}

gibbs_samps <- uni_pareto_gibbs(S = 100000, r, m, k, alpha, beta, burn_in=2000)
```

Exercise 2

Here we plot the inferred R^2 posterior marginal against the true R^2. It looks like we are doing an excellent job of estimating the mean and the shape.

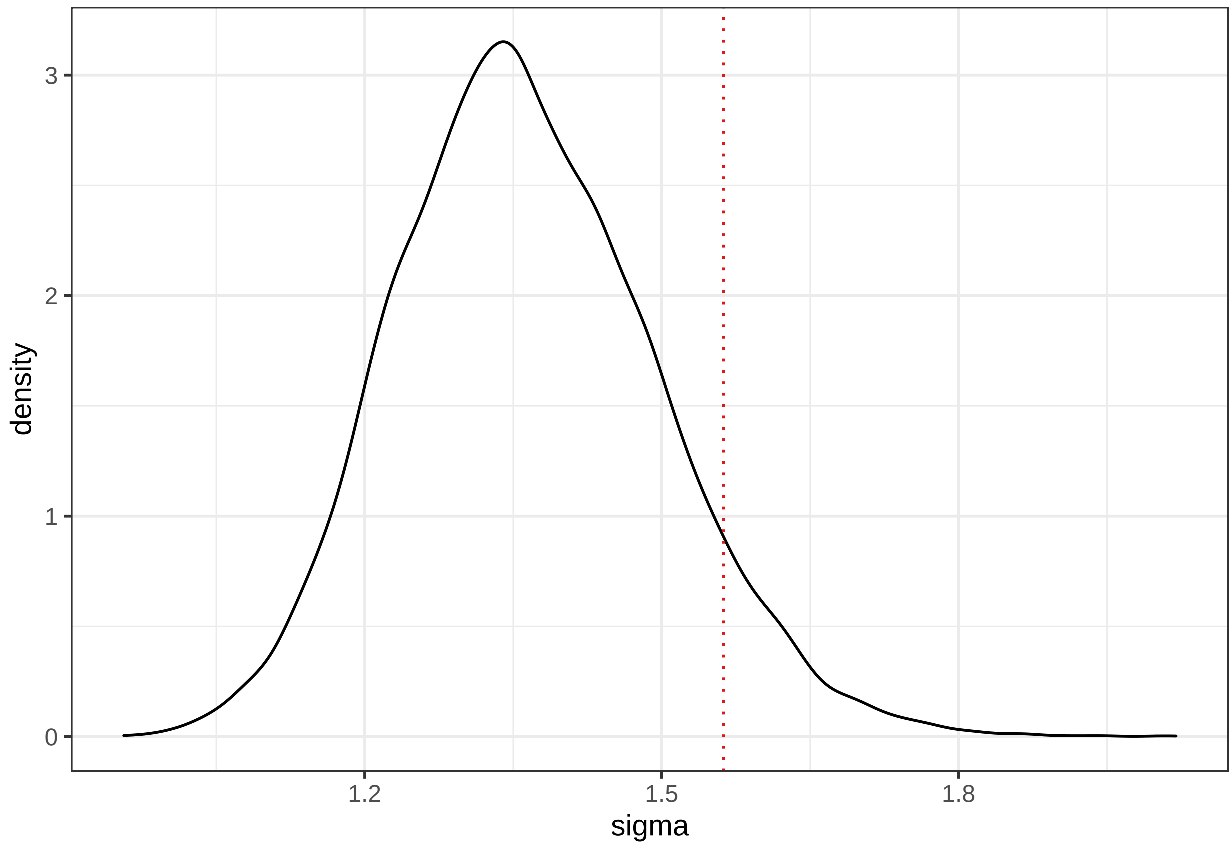
```r
ggplot() +
  geom_density(data = data.frame(R = gibbs_samps$R), aes(x=R)) +
  geom_vline(xintercept = true_Rsquared, color='#da1a1a', linetype="dotted")
```



Exercise 3

Here we plot the inferred σ^2 posterior marginal against the true $sigma^2$. On average, we are underestimating the noise term here.

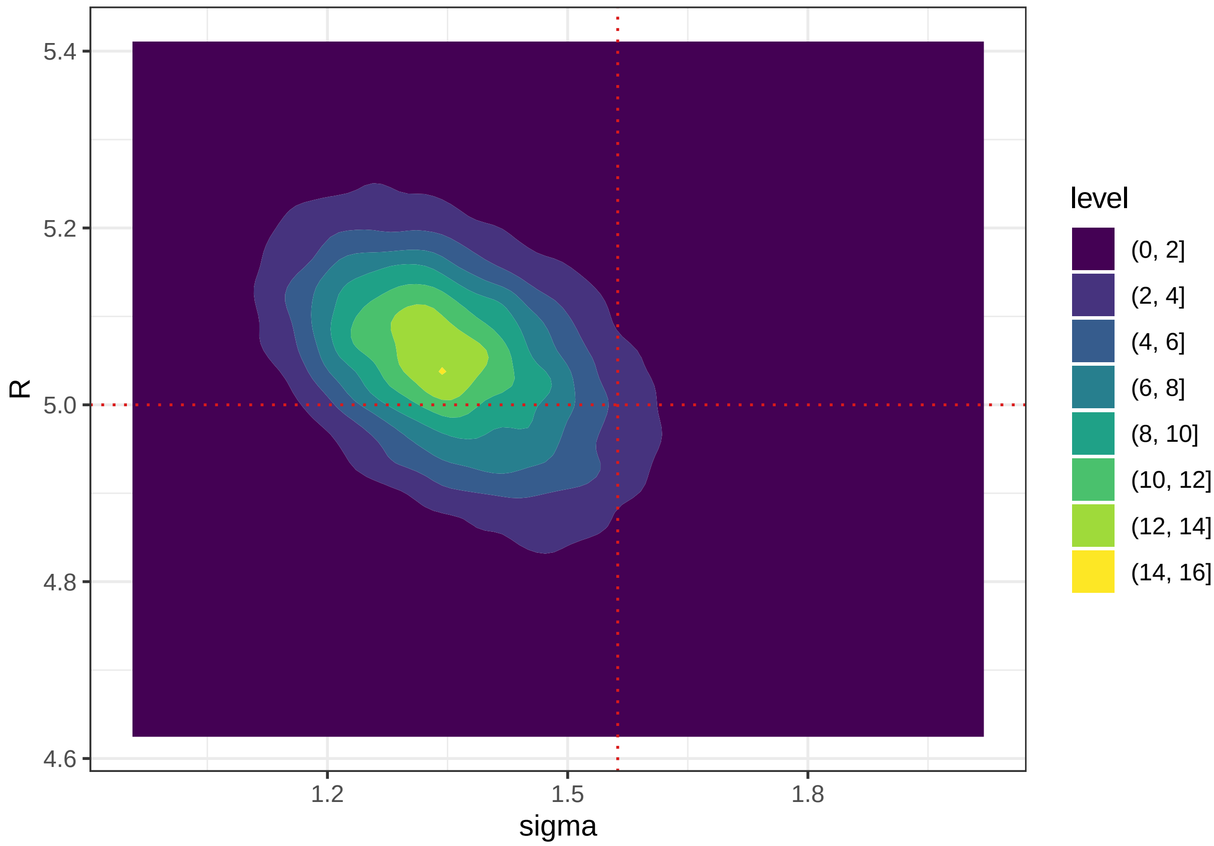
```
ggplot() +
  geom_density(data = data.frame(sigma = gibbs_samps$sigma^2), aes(x=sigma)) +
  geom_vline(xintercept = true_sigma^2, color='#da1a1a', linetype="dotted")
```



Exercise 4

Furthermore, here we do a 2D contour plot between (R^2, σ^2) and compare the distribution to the true parameter. Matching with the previous comments that we are underestimating the variances.

```
ggplot() +
  geom_density2d_filled(data = data.frame(R = gibbs_samps$R, sigma = gibbs_samps$sigma^2), aes(sigma, R)
  geom_vline(xintercept = true_sigma^2, color='#da1a1a', linetype="dotted") +
  geom_hline(yintercept = true_Rsquared, color='#da1a1a', linetype="dotted")
```



5.4
5.2
5.0
4.8
4.6
R
1.2
1.5
1.8
sigma
level
(0, 2]
(2, 4]
(4, 6]
(6, 8]
(8, 10]
(10, 12]
(12, 14]
(14, 16]