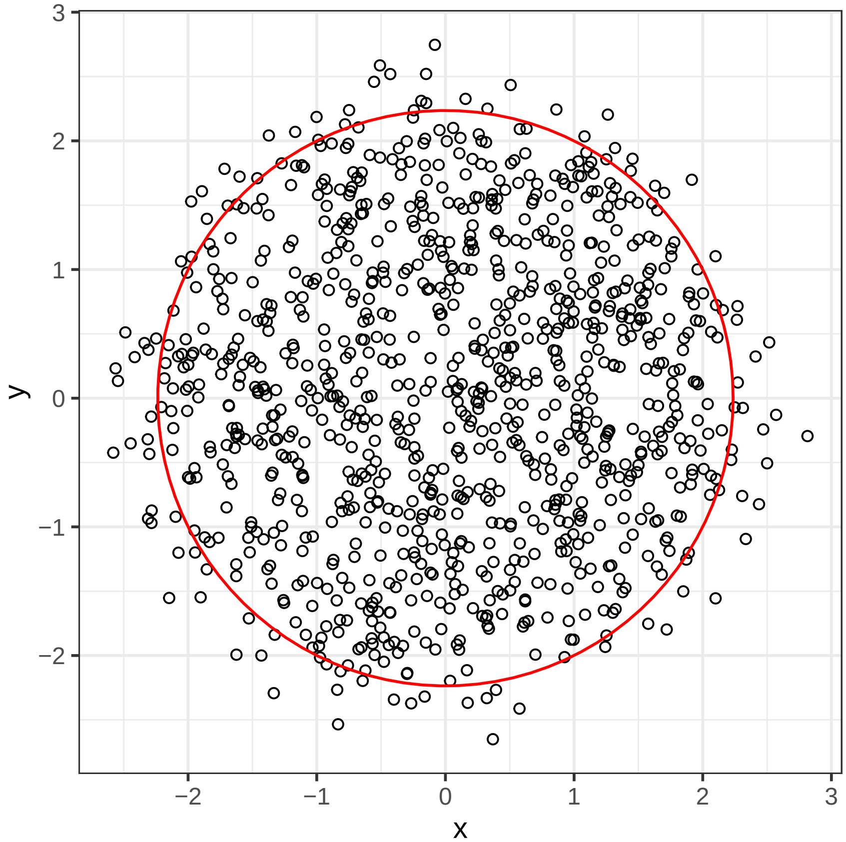
Lab5

Bingying Liu

2/17/2020

Exercise 1

```r
#
n <- 1000
true_Rsquared <- 5
true_sigma <- 1.25
#
u <- runif(n, 0, true_Rsquared)
r <- u + rnorm(n, sd = true_sigma)
theta <- runif(n, 0, 2*pi)
#
ggplot2::ggplot() +
  geom_point(data = data.frame(x = sign(r)*sqrt(abs(r))*cos(theta), y = sign(r)*sqrt(abs(r))*sin(theta)
             aes(x = x, y = y), shape = 1) +
  geom_path(data = data.frame(R = true_Rsquared) %>%
                              plyr::ddply(.(R), function(d){
                                data.frame(x = sqrt(d$R)*cos(seq(0, 2*pi, length.out = 100)),
                                           y = sqrt(d$R)*sin(seq(0, 2*pi, length.out = 100)))
                              }),
                       aes(x = x, y = y), alpha = 1, colour = "red") +
  coord_fixed()
```



Exercise 1:

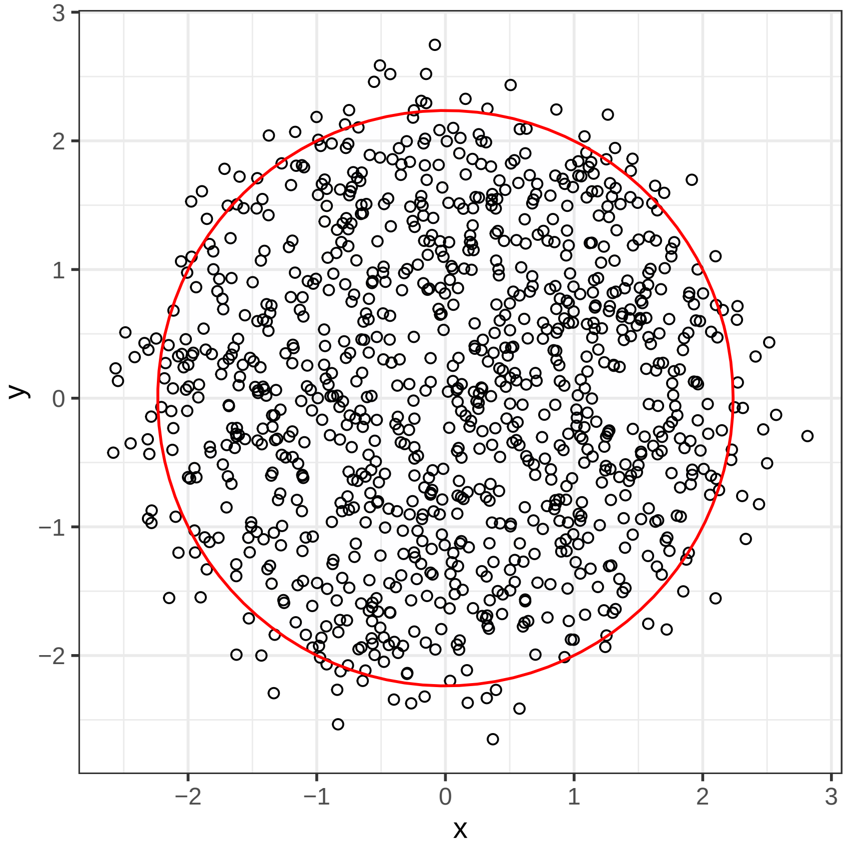
```
# hyper-parameters
m <- 3
k <- 1
alpha <- 5/2
beta <- 5/2

#
rpareto <- function(m, k, trunc = NULL){
  p <- m*(1 - runif(1))^(-1/k)
  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
  n <- length(Rsq)
```

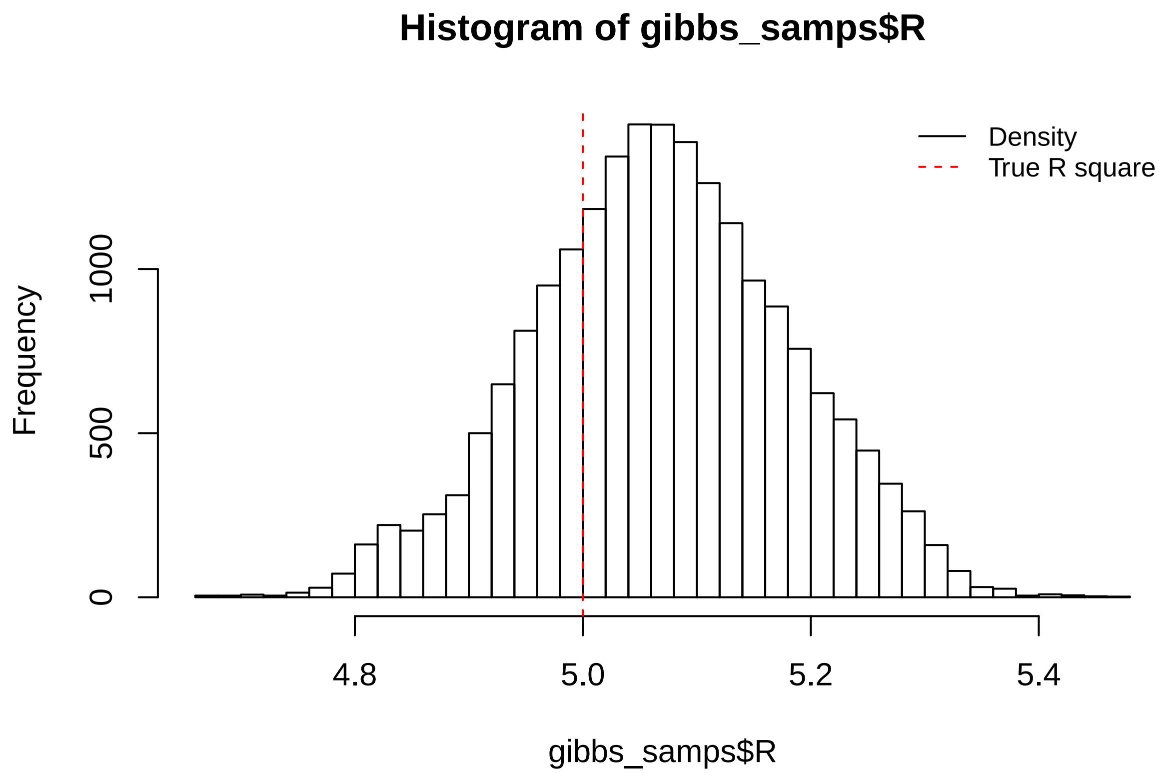
```r
  R <- rep(1, S)
  U <- matrix(0, nrow = S, ncol = n)
  U[1, ] <- runif(n, 0, R)
  sigma <- rep(1, S)
  #
  U_curr <- U[1, ]
  R_curr <- R[1]
  sigma_curr <- sigma[1]
  for(s in 1:S){
    # Sample from full conditional of the inner radius
    R_curr <- rpareto(max(c(U_curr, m)), k + n)
    R[s] <- R_curr
    # Sample from full conditional of U values
    U_curr <- truncnorm::rtruncnorm(n, a = 0, b = R_curr, mean = Rsq, sd = sigma_curr)
    U[s, ] <- U_curr
    # Sample from full conditional of sigma
    precision <- rgamma(1, n/2+alpha, 0.5*sum((U_curr-Rsq)**2)+beta)
    sigma_curr <-  sqrt(1/precision) #complete this line
    sigma[s] <- sigma_curr
  }
  return(list(R = R[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)
```

```r
ggplot2::ggplot() +
  geom_point(data = data.frame(x = sign(r)*sqrt(abs(r))*cos(theta), y = sign(r)*sqrt(abs(r))*sin(theta)
             aes(x = x, y = y), shape = 1) +
  geom_path(data = data.frame(R = gibbs_samps$R) %>%
                              plyr::ddply(.(R), function(d){
                                data.frame(x = sqrt(d$R)*cos(seq(0, 2*pi, length.out = 100)),
                                           y = sqrt(d$R)*sin(seq(0, 2*pi, length.out = 100)))
                              }),
                       aes(x = x, y = y), alpha = 0.005, colour = "blue") +
  geom_path(data = data.frame(R = true_Rsquared) %>%
                              plyr::ddply(.(R), function(d){
                                data.frame(x = sqrt(d$R)*cos(seq(0, 2*pi, length.out = 100)),
                                           y = sqrt(d$R)*sin(seq(0, 2*pi, length.out = 100)))
                              }),
                       aes(x = x, y = y), alpha = 1, colour = "red") +
  coord_fixed()
```



Exercise 2

```
hist(gibbs_samps$R, breaks = 50)
abline(v = 5, col=c("red"), lty=2, lwd=c(1))
legend("topright", legend=c("Density", "True R square"), col=c("black", "red"), lty=1:2, cex=0.8, bg="t
```




```
mean(gibbs_samps$R)
```

```
## [1] 5.068893
```

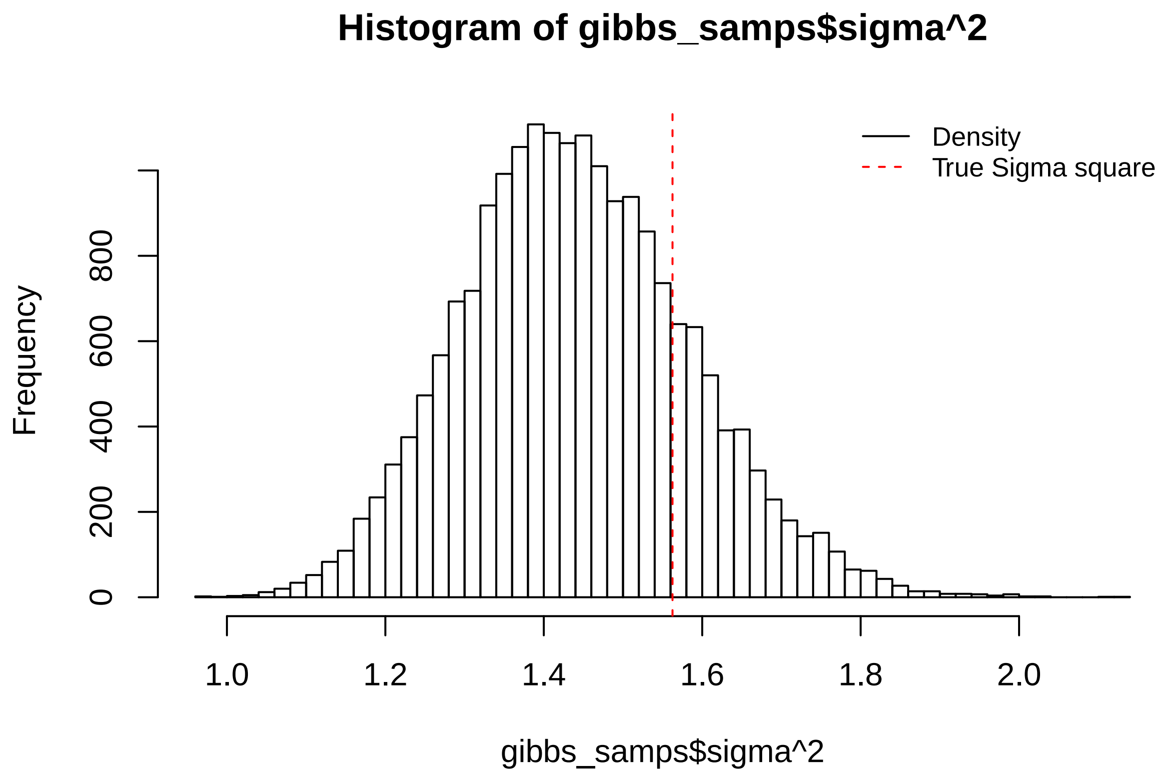
```
true_Rsquared
```

```
## [1] 5
```

The true value of R^2 is 5. The mean of the marginal posterior density of R^2 is close to the true value.

Exercise 3

```
hist(gibbs_samps$sigma**2, breaks = 50)
abline(v = 1.25^2, col=c("red"), lty=2, lwd=c(1))
legend("topright", legend=c("Density", "True Sigma square"), col=c("black", "red"), lty=1:2, cex=0.8, bg
```




```
mean(gibbs_samps$sigma**2)
```

```
## [1] 1.441534
```

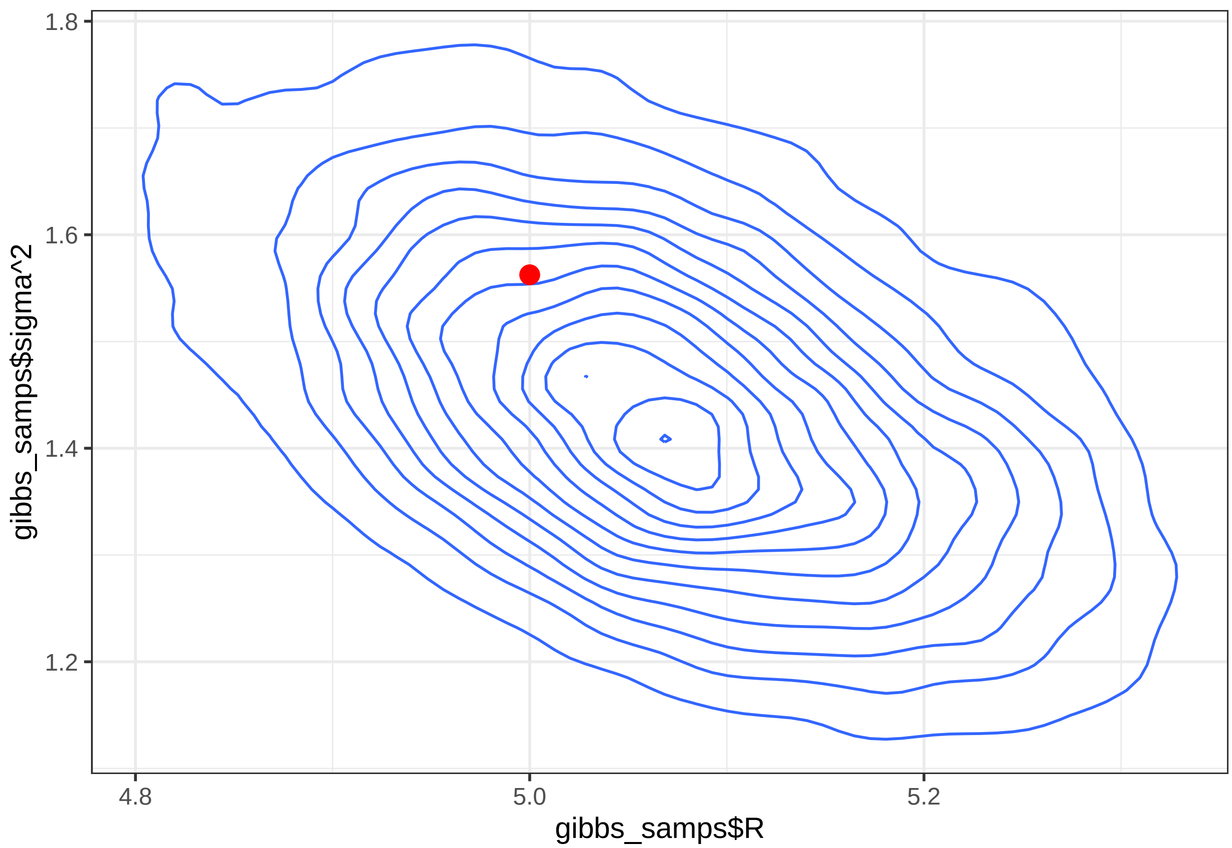
```
true_sigma**2
```

```
## [1] 1.5625
```

The true value of σ^2 is 1.5625. The mean of the marginal posterior density of σ^2 is about 1.7.

Exercise 4

```
p <- qplot(gibbs_samps$R, gibbs_samps$sigma**2, geom="density2d") + geom_point(aes(true_Rsquared, true_s
cowplot::plot_grid(p)
```



The truth on the contour plot of posterior bivariate density lies in the high density area, which means the sampler could simulate very well.