STATS 600 HW2

October 3, 2023

Contents

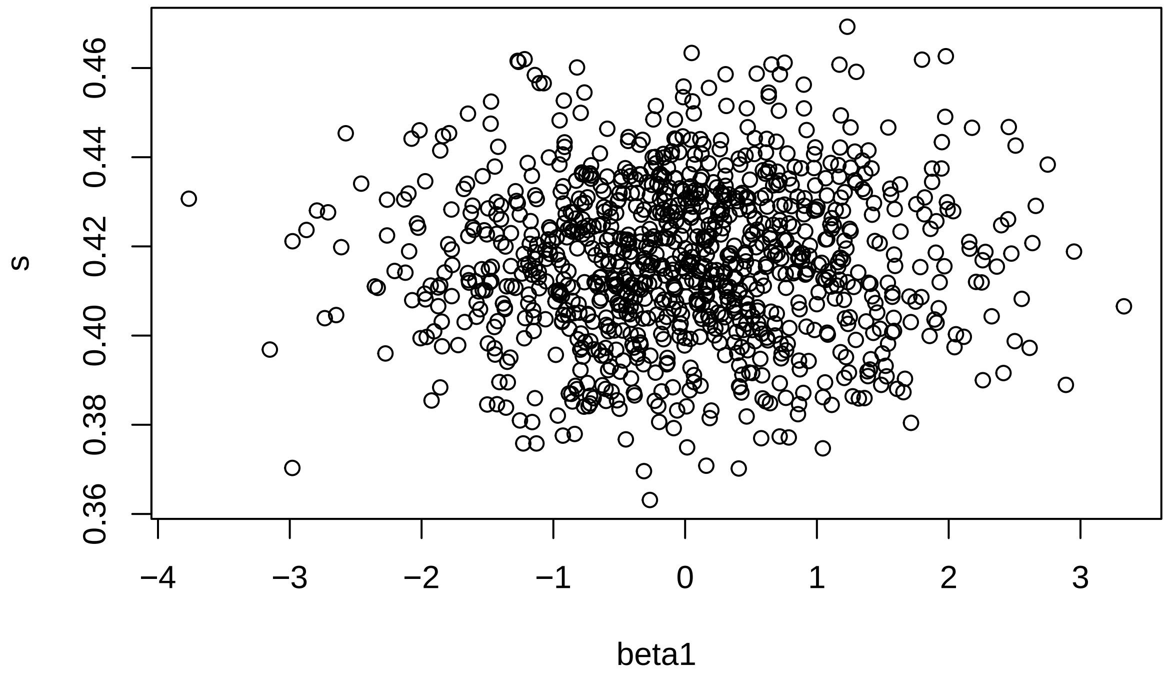
```r
# Generating Design Matrix X with size n = 100 and two covariates d = 2
n = 100
d = 5
# In order to truly ensure that we have rank d, we can just use an identity matrix
# and append the rest of the matrix with 0 to form X. We have a large vertical matrix
# because the matrix is n x d.
X = rbind(diag(d), matrix(0, n-d, d))
# Instead of randomly generated beta we can just make it fixed.
# beta must be d x 1
beta = seq(1, d)


# This is for sigma, not sigma^2
epsilon = rnorm(d, mean = 0, sd = 2)
Y = X%*%beta + epsilon
beta_hat = solve(t(X)%*%X) %*% t(X) %*% Y

Yhat = X%*%beta_hat
RSS = norm(Y - Yhat, type = "2")
sigma_hat = sqrt(RSS / (n-d))
se_hat = sigma_hat * sqrt(t(diag(d))%*%solve(t(X)%*%X)%*%diag(d))

# Extracting the diagonal of the matrix se_hat and doing entry-wide division
# from beta_hat and diagonal elements
t = beta_hat / diag(se_hat) # Each entry of t is t_j
```

```r
library(ExtDist)
```

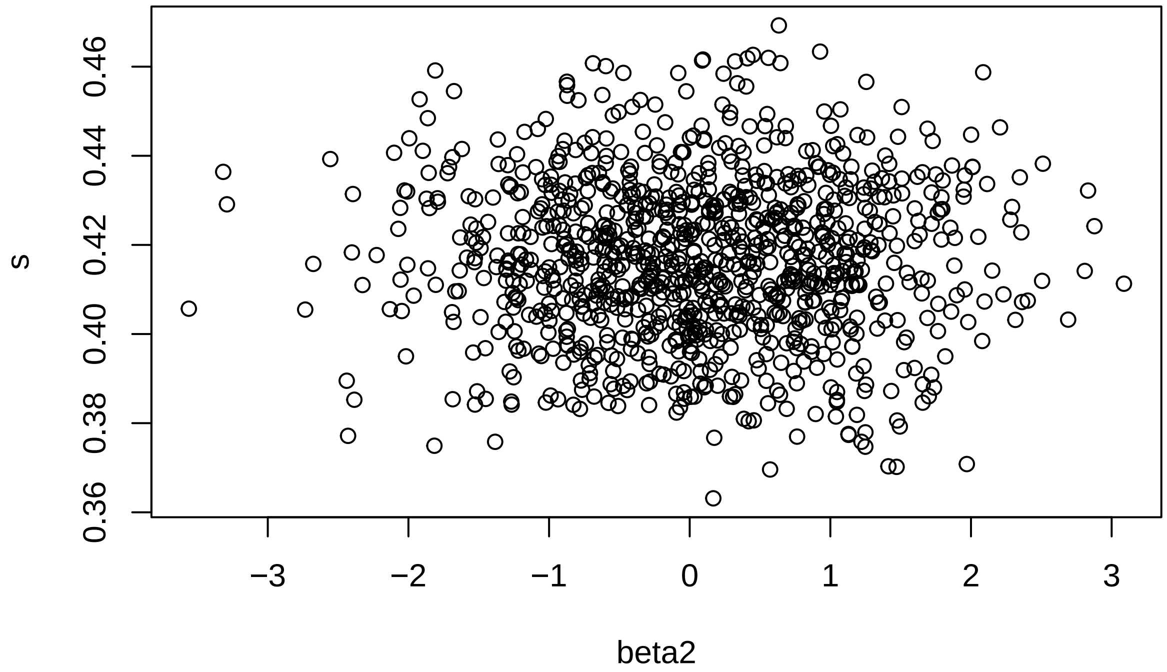
```
##
## Attaching package: ’ExtDist’

## The following object is masked from ’package:stats’:
##
##     BIC
```

```r
# Generating Design Matrix X with size n = 100 and two covariates
n = 100
p = 2
set.seed(1)
```

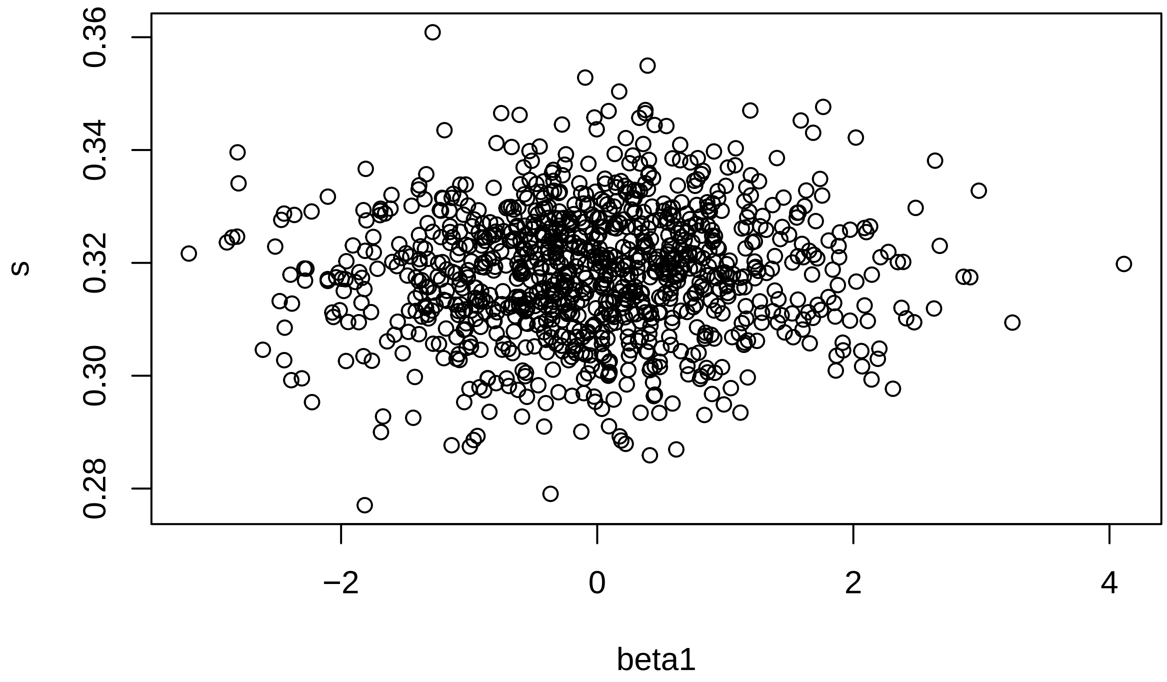
```
beta1 <- list()
beta2 <- list()
s <- list()

for (i in 1:1000) {
  X = matrix(rnorm(n*p, 0, 1), n, p)
  beta = rnorm(p, 0, 1)

  epsilon = rLaplace(n, mu = 1, sigma = 2)
  Y = X%*%beta + epsilon
  beta_hat = solve(t(X)%*%X) %*% t(X) %*% Y
  beta_hat

  Yhat = X%*%beta_hat
  RSS = norm(Y - Yhat, type = "2")
  sigma_hat = sqrt(RSS / (n-p))
  sigma_hat

  beta1[i] <- beta_hat[1]
  beta2[i] <- beta_hat[2]
  s[i] <-sigma_hat
}

plot(beta1, s)
```



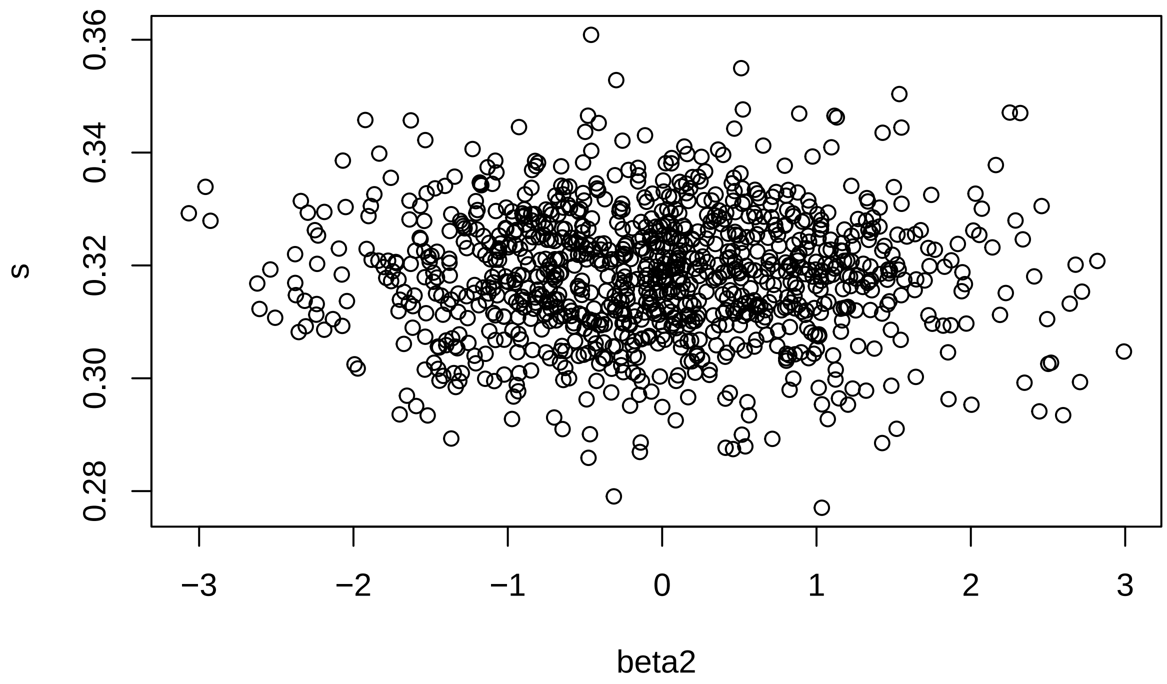
```r
plot(beta2, s)
```



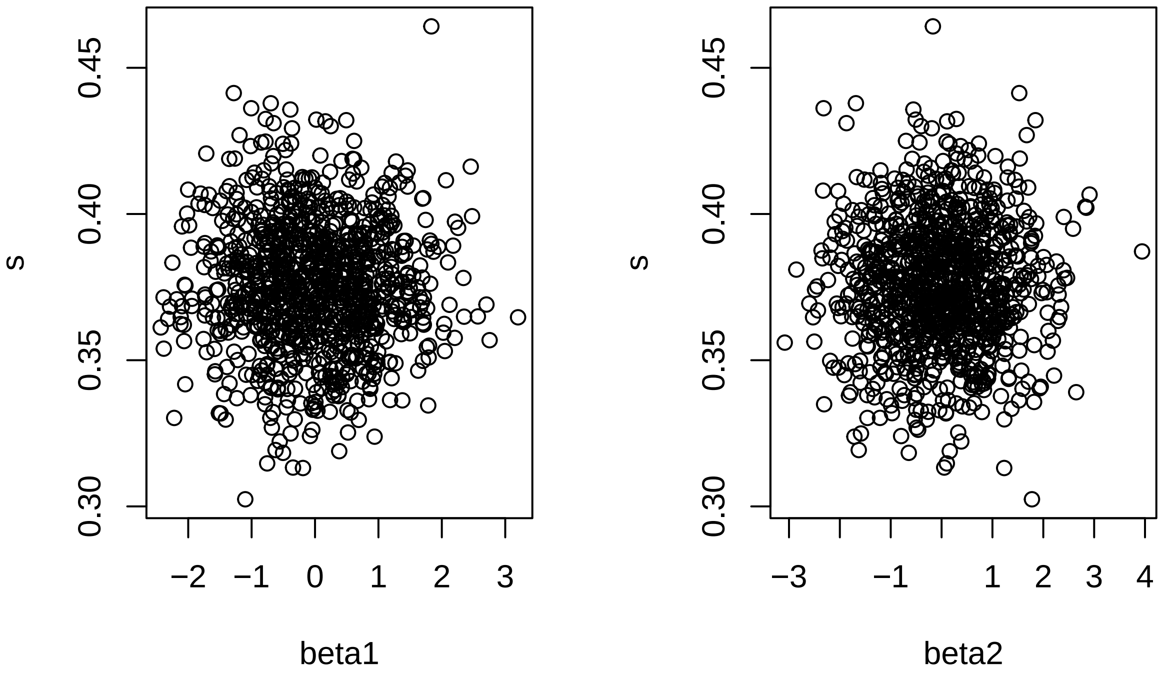
```r
library(ExtDist)
# Generating Design Matrix X with size n = 100 and two covariates
n = 100
p = 2
set.seed(1)

beta1 <- list()
beta2 <- list()
s <- list()

for (i in 1:1000) {
  X = matrix(rnorm(n*p, 0, 1), n, p)
  beta = rnorm(p, 0, 1)

  epsilon = rt(n, df = n-p)
  Y = X%*%beta + epsilon
  beta_hat = solve(t(X)%*%X) %*% t(X) %*% Y
  beta_hat

  Yhat = X%*%beta_hat
  RSS = norm(Y - Yhat, type = "2")
  sigma_hat = sqrt(RSS / (n-p))
  sigma_hat
  beta1[i] <- beta_hat[1]
  beta2[i] <- beta_hat[2]
  s[i] <-sigma_hat
}

plot(beta1, s)
```



```
plot(beta2, s)
```



```
library(ExtDist)
# Generating Design Matrix X with size n = 100 and two covariates
n = 100
p = 2
set.seed(1)

beta1 <- list()
beta2 <- list()
s <- list()

for (i in 1:1000) {
  X = matrix(rnorm(n*p, 0, 1), n, p)
  beta = rnorm(p, 0, 1)

  epsilon = rexp(n, rate = 1)
  Y = X%*%beta + epsilon
  beta_hat = solve(t(X)%*%X) %*% t(X) %*% Y
  beta_hat

  Yhat = X%*%beta_hat
  RSS = norm(Y - Yhat, type = "2")
  sigma_hat = sqrt(RSS / (n-p))
  sigma_hat

  beta1[i] <- beta_hat[1]
  beta2[i] <- beta_hat[2]
  s[i] <-sigma_hat
}

par(mfrow = c(1, 2))
plot(beta1, s)
plot(beta2, s)
```


s
beta1
s
beta2