

PS1 Mark Scheme

Professor Xavi, Linda Xie

Sept 13, 2019

0. preliminaries AND comments [6]

```
##clear memory [1]
rm(list=ls()); gc(); graphics.off()

##          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
## Ncells 450685 24.1      972414   52          NA   630514 33.7
## Vcells 877151  6.7      8388608   64        16384 1767157 13.5

##comments and submission [5]
```

1. Matrix X_data and column vector betas [8]

1.1 Creating matrix X_data [6]

```
##number of columns and rows [2]
num_cols      = 12
num_rows      = 1000
##generate random numbers [2]
X_data        = matrix(runif(num_cols*num_rows, min=1, max=15), ncol=num_cols, nrow=num_rows)
##round number to 3 decimals [1]
X_data        = round(X_data, 3)
##name columns [1]
colnames(X_data) = paste('X_data', 1:ncol(X_data), sep='_')
```

1.2 Creating column vector betas [2]

```
##generate random numbers [1]
betas         = sample(1:5, size=num_cols, replace = TRUE)
##name betas
names(betas) = paste('beta_', 1:length(betas), sep='')
##convert vector to column vector [1]
betas         = cbind(betas)
```

2. Creating Y_data and noise [5]

```
##create noise vector [2]
vector_noise = cbind(rnorm(num_rows, mean = 0, sd = 4))
##calculate Y_data [3]
Y_data       = X_data%*%betas + vector_noise
```

3. Beta_hat [3]

```
##use formula to create beta_hat [3]
beta_hat = solve(t(X_data)%*%X_data)%*%t(X_data)%*%Y_data
```

4. Estimation error [4]

```
##create estimation error [2]
error_analysis = cbind(betas,beta_hat,difference = abs(betas-beta_hat))
##name columns [1]
colnames(error_analysis) = c('Beta True','Beta Estimated','Estimation Error')
##find max error [1]
max(error_analysis[, 'Estimation Error'])

## [1] 0.02770203
```

5. Sample size increment [2]

```
##record max error for each sample size [2]
errors = rep(NA, 20)
names(errors) = paste(1:20, 'K', sep='_')
errors[1] = max(error_analysis[, 'Estimation Error'])
```

6. forloop (optional) or written explanation [2]

```
for(i in 2:20)
{
  num_cols = 12
  num_rows = i%%1000
  X_data = matrix(runif(num_cols*num_rows, min=1, max=15), ncol=num_cols, nrow=num_rows)
  X_data = round(X_data, 3)
  colnames(X_data) = paste('X_data', 1:ncol(X_data), sep='_')
  betas = sample(1:5, size=num_cols, replace = TRUE)
  names(betas) = paste('beta_', 1:length(betas), sep='')
  betas = cbind(betas)
  vector_noise = cbind(rnorm(num_rows, mean = 0, sd = 4))
  Y_data = X_data%%betas + vector_noise
  beta_hat = solve(t(X_data)%%X_data)%%t(X_data)%%Y_data
  error_analysis = cbind(betas,beta_hat,difference = abs(betas-beta_hat))
  colnames(error_analysis) = c('Beta True','Beta Estimated','Estimation Error')
  max(error_analysis[, 'Estimation Error'])
  errors[i] = max(error_analysis[, 'Estimation Error'])
}

barplot(errors)
```

