Stat 230: Linear Models
Homework 2
Professor Ding
Lev Golod

Question 6: Part 1

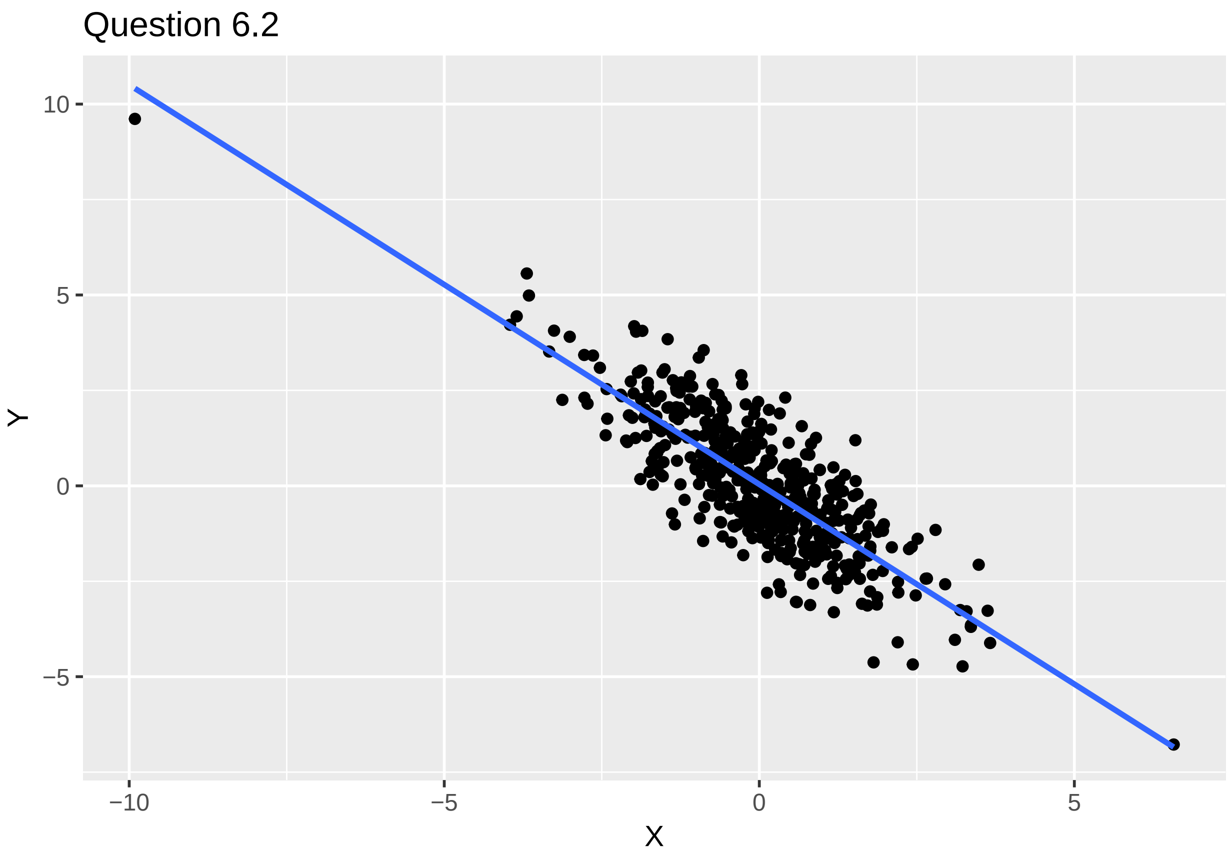
```
j <- 1000
# j <- 30
n <- 500
r <- 10
e <- rnorm(n = n)
X <- rt(n = n, df = 6)
Y <- as.matrix(-1 * X + e)
data1 <- data.frame(cbind(Y, X))
```

Question 6: Part 2

```
reg1 <- lm(Y ~ X, data = data1)
reg1
```

```
##
## Call:
## lm(formula = Y ~ X, data = data1)
##
## Coefficients:
## (Intercept)            X
##     0.03854     -1.04668
```

```
myplot1 <- ggplot(data1, aes(x = X, y = Y)) +
  geom_point() +
  ggtitle("Question 6.2") +
  geom_smooth(method = 'lm', se = FALSE)
myplot1
```



Question 6: Part 3

```
# To get an intercept we need to add a column of 1s to X
X <- cbind(rep(1, n), X)
colnames(X) <- NULL

# Pi vector - sampling probability weights
Pi <- rep(1/n,n)

# True beta: (0, -1)
truebeta <- c(0, -1)

# # function to calculate MSE
# mse <- function(y, yhat){
#   if (length(y) != length(yhat)) stop("y and yhat must have same length")
#   length(y)**-1 * sum( (y-yhat)**2 )
# }

# a function that does 1 iteration of Weighted Leveraging
WL_iter1 <- function(){

  # identify the sample indices and weights
  i <- sample.int(n, r, prob = Pi, replace = TRUE)
  wt <- Pi[i]

  # fit the model, identify coefficients and MSE
```

```r
  fit <- lm(y ~ x, data = data.frame(y = Y[i,], x = X[i,2]),
            weights = wt)
  coeffs <- fit$coefficients
  # mse_i <- mse(coeffs, truebeta)

  # return the result as a 1-row matrix
  # result <- cbind(t(coeffs), mse_i)
  result <- cbind(t(coeffs))
  colnames(result) <- c('beta0', 'beta1')
  return(result)
}

results <- replicate(j, WL_iter1())
results <- t(results[,,])


## report mean coefficients
apply(results, 2, mean)
```

```
##       beta0       beta1
##  0.03114914 -1.05136567
```

```r
## report MSE
mse <- c(
(1/n)*sum((results[,1] - 0)**2),
(1/n)*sum((results[,2] - -1)**2)
)
names(mse) <- c('MSE_beta0', 'MSE_beta1')
mse
```

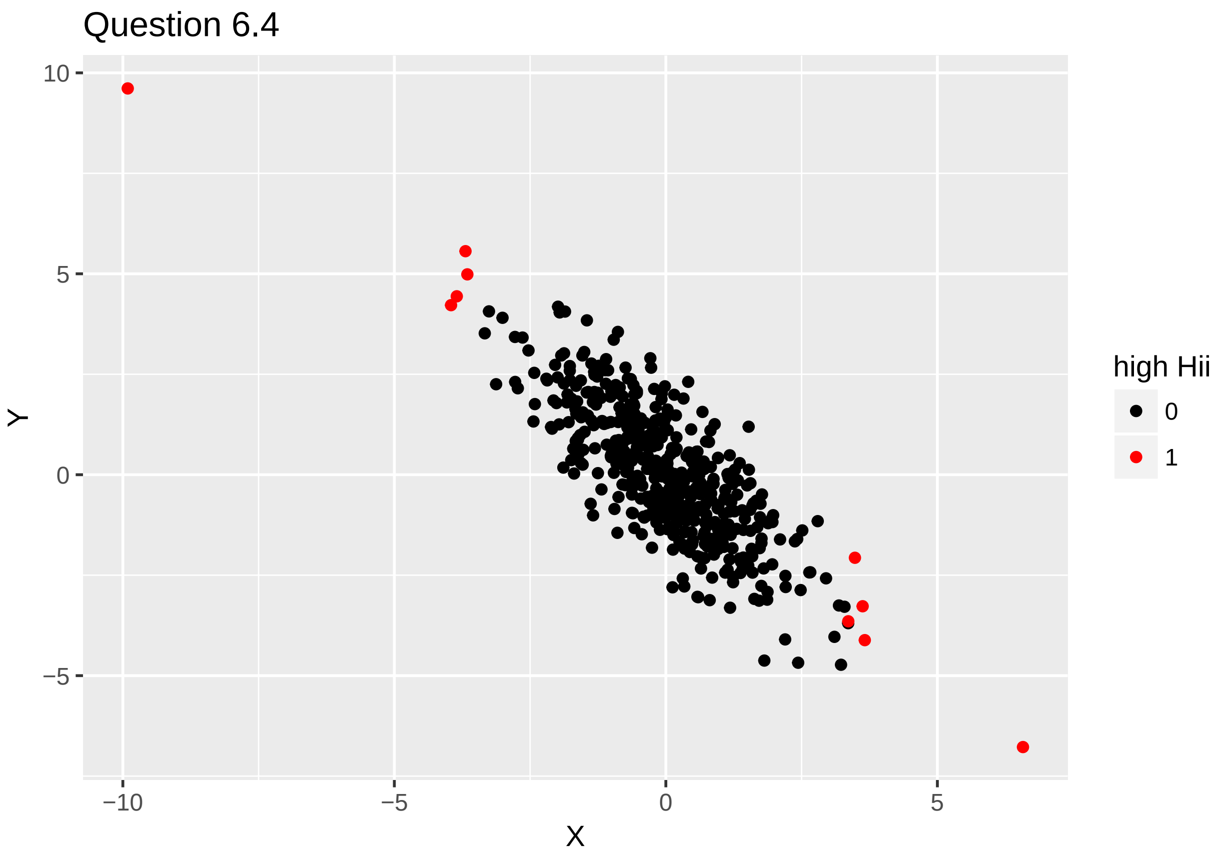
```
## MSE_beta0 MSE_beta1
## 0.2415014 0.2027846
```

Question 6: Part 4

```r
H = X %*% solve(t(X) %*% X) %*% t(X)
Hii = diag(H)
names(Hii) <- 1:n

# which are the r highest Hii values?
high = as.numeric(names(sort(Hii, decreasing = TRUE)[1:10]))
data2 <- data.frame(cbind(Y, X[,2], 1:n %in% high))
names(data2) <- c("Y", "X", "highHii")

myplot2 <- ggplot(data2, aes(x = X, y = Y, color = factor(highHii))) +
  geom_point() +
  scale_color_manual(name = "high Hii", values = c("black", "red")) +
  ggtitle("Question 6.4")
myplot2
```



Question 6: Part 5

```
rm(results)
Pi = Hii / sum(Hii)

# a function that does 1 iteration of Weighted Leveraging
WL_iter2 <- function(q){

  # identify the sample indices
  i <- sample.int(n, r, prob = Pi, replace = TRUE)

  wt <- Pi[i]

  # fit the model, identify coefficients and MSE
  fit2 <- lm(y ~ x, data = data.frame(y = Y[i,], x = X[i,2]),
          weights = wt**(-0.5))
  # mse(fit1) - mse(fit2)

  coeffs <- fit2$coefficients
  # mse_i <- mse(coeffs, truebeta)

  # return the result as a 1-row matrix
  # result <- cbind(t(coeffs), mse_i)
  result <- cbind(t(coeffs))
  colnames(result) <- c('beta0', 'beta1')
  return(result)
```

```r
}

results <- replicate(j, WL_iter2())
results <- t(results[,,])

## report mean coefficients and MSEs
apply(results, 2, mean)
```

```
##       beta0       beta1
##  0.05298914 -1.03132197
```

```r
## report MSE
mse <- c(
(1/n)*sum((results[,1] - 0)**2),
(1/n)*sum((results[,2] - -1)**2)
)
names(mse) <- c('MSE_beta0', 'MSE_beta1')
mse
```

```
##  MSE_beta0  MSE_beta1
## 0.25293993 0.05627262
```

Comment: As discussed in class, using the Hat Matrix to weight the data leads to superior regression results. We see this because the MSE associated with beta_1 is smaller.