Assignment 2

Group 11

Chang Zhou 01983512, Qian Zhang 01939418, Yutong Zheng 01895402

2021/5/17

Question 1:

(a).

$$\beta^* = (X^T X + \lambda I)^{-1} X^T y$$

$X^T X$ could be written as $\sum_{i=1}^{N} z_i^T z_i$. so we don't need to calculate $X^T X$ directly.

For example. if we split X to $[X_1 : X_2]$, and Y to $[Y_1 : Y_2]$

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{bmatrix} \quad X^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & x_{21} & \cdots & x_{N1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1d} & x_{2d} & \cdots & x_{Nd} \end{bmatrix} \quad X^T X = \begin{bmatrix} (1)(1) + \cdots (1)(1), & \cdots & x_{1d}(1) + \cdots x_{Nd}(1) \\ \vdots & \ddots & \vdots \\ x_{1d}(1) + x_{2d}(1) + \cdots x_{Nd}(1) & \cdots & x_{1d}(x_{1d}) + \cdots x_{Nd}(x_{Nd}) \end{bmatrix}$$

$$X \in \mathbb{R}^{N \times (d+1)} \qquad X^T \in \mathbb{R}^{(d+1) \times N} \qquad X^T X \in \mathbb{R}^{(d+1) \times (d+1)}$$

$$X_1 = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{\frac{N}{2}1} & x_{\frac{N}{2}2} & \cdots & x_{\frac{N}{2}d} \end{bmatrix} \quad X_1^T = \begin{bmatrix} 1 & \cdots & 1 \\ x_{11} & \cdots & x_{\frac{N}{2}1} \\ \vdots & \ddots & \vdots \\ x_{1d} & \cdots & x_{\frac{N}{2}d} \end{bmatrix} \quad X_1^T X = \begin{bmatrix} (1)(1) + \cdots (1)(1) & \cdots & x_{1d}(1) + \cdots x_{\frac{N}{2}d}(1) \\ \vdots & \ddots & \vdots \\ x_{1d}(1) + x_{2d}(1) + \cdots x_{\frac{N}{2}d}(1) & \cdots & x_{1d}(x_{1d}) + \cdots x_{Nd}(x_{Nd}) \end{bmatrix}$$

$$X_1 \in \mathbb{R}^{\frac{N}{2} \times (d+1)} \qquad X_1^T \in \mathbb{R}^{(d+1) \times \frac{N}{2}} \qquad X_1^T X_1 \in \mathbb{R}^{(d+1) \times (d+1)}$$

$$X_2 = \begin{bmatrix} 1 & x_{(\frac{N}{2}+1)1} & \cdots & x_{(\frac{N}{2}+1)d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{bmatrix} \quad X_2^T = \begin{bmatrix} 1 & \cdots & 1 \\ x_{(\frac{N}{2}+1)1} & \cdots & x_{N1} \\ \vdots & \ddots & \vdots \\ x_{(\frac{N}{2}+1)d} & \cdots & x_{Nd} \end{bmatrix} \quad X_2^T X = \begin{bmatrix} (1)(1) + \cdots (1)(1) & \cdots & x_{(\frac{N}{2}+1)d}(1) + \cdots x_{Nd}(1) \\ \vdots & \ddots & \vdots \\ x_{(\frac{N}{2}+1)d}(1) + \cdots x_{Nd}(1) & \cdots & x_{(\frac{N}{2}+1)d}\, x_{(\frac{N}{2}+1)d} + \cdots x_{Nd} x_{Nd} \end{bmatrix}$$

$$X_2 \in \mathbb{R}^{\frac{N}{2} \times (d+1)} \qquad X_2^T \in \mathbb{R}^{(d+1) \times \frac{N}{2}} \qquad X_2^T X_2 \in \mathbb{R}^{(d+1) \times (d+1)}$$

We could see that $X^T X \in X_1^T X_1 + X_2^T X_2$.

Similarity, $X^T y \in X_1^T Y_1 + X_2^T Y_2$. where $Y = [Y_1 : Y_2]$

Calculating β^*:

First. we load the first row X_i of X dataset and the row Y_i of the Y dataset.

Then, we compute $X_1^T \cdot X_1$ and $X_1^T \cdot Y_1$. since $X_1 \in \mathbb{R}^{1\times(d+1)}$, $Y_1 \in \mathbb{R}^{1\times1}$, and dimension d of the datapoint is not large. it's doesn't require much space to calculate and store $X_1^T \cdot X_1$ and $X_1^T \cdot Y_1$, where $X_1^T \cdot X_1 \in^{(d+1)\times(d+1)}$

Then. we load the second row of X and Y dataset. compute $X_2^T \cdot X_2$ and $X_2^T Y_2$, and add the results to $X_1^T \cdot X_1$ and $X_1^T \cdot Y_1$. We repeat this step until whole datasets X and Y are read. While the runtime of this step is still very large. the required memory space is not large.

Last. after we compute the final $X^T X$ and $X^T Y$. $\beta^* = (X^T X + \lambda I)^{-1} X^T Y$

(b). As describled in part a. we could calalate $X^T X$ and $X^T Y$ be summing $X_i^T X_i$ and $Y_i^T Y_i$. where X_i and Y_i are the ith row of dataset X and Y. since $X_i^T X_i \in \mathbb{R}^{(d+1)(d+1)}$, each time we only need to keep $O(d^2)$ numbers in the database.

For this question. similiarly, we could calculate $X^T X$ and $X^T Y$ using all the previous data $\{X_1, \cdots, X_k\}$. For each time a new datapoint X_{k+1} arrives. we could calculate $X_{k+1}^T X_{k+1}$ and $X_{k+1}^T Y_{k+1}$. and add them to the $X^T X$ and $X^T Y$ that are stored in the database. and use the new values to calculate the new β_k^*.

(c)

Suppose when $n=1$, $X^TX = x_1^Tx_1$, then $\mu_1 = (\lambda I + x_1^Tx_1)^{-1} = (\lambda I)^{-1} - \frac{(\lambda I)^{-1}x_1^Tx_1(\lambda I)^{-1}}{1+x_1(\lambda I)^{-1}x_1^T}$.
Since λI is just a diagonal matrix, $(\lambda I)^{-1}$ is obtained by taking the inverse of its diagonal element λ. As for $(\lambda I)^{-1}x_1^T$, it's the multiplication of a $(d+1)\times(d+1)$ matrix with a $(d+1)\times 1$ vector, so it only takes $O(d^2)$ time.
Similarly for the rest, the calculation of $\frac{(\lambda I)^{-1}x_1^Tx_1(\lambda I)^{-1}}{1+x_1(\lambda I)^{-1}x_1^T}$ takes only $O(d^2)$ time. Therefore, we can store the result μ_1 to simplify further calculation.
For example when $n=2$, $X^TX = x_1^Tx_1 + x_2^Tx_2$, then
$\mu_2 = (\lambda I + x_1^Tx_1 + x_2^Tx_2)^{-1} = [(\lambda I + x_1^Tx_1) + x_2^Tx_2]^{-1} = \mu_1 - \frac{\mu_1 x_2^T x_2 \mu_1}{1+x_2\mu_1 x_2^T}$.
We've already calculated μ_1 and the multiplication in the second part takes only $O(d^2)$ time, so we can update μ_1 to μ_2 for 3rd round.
Iteratively, we only have to calculate $\frac{\mu_i x_{i+1}^T x_{i+1} \mu_i}{1+x_{i+1}\mu_i x_{i+1}}$ in the $n=i+1$ round to get the inverse of matrix $X^TX+\lambda I$.
Hence, it only takes $O(d^2)$ time to compute $(X^TX+\lambda I)^{-1}$ in each round and multiply it with X^Ty ($(d+1)\times 1$ vector), which also only takes $O(d^2)$ time.

Question 2

(a) The proportion of parked cars in violation of the meters in the population is approximated by the number of "violation" cars in randomly selected blocks divided by the total number of parking cars in those blocks.

(b) If we can assume all the parking cars are i.i.d. samples, or differences between blocks are really small and all cars are parking at random among all blocks, then we can regard them as Bernoulli trials with true p. Therefore $\frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n z_i}$ is a reasonable estimate of p when simple random sampling is inefficient and expensive.

(c) The usual formula may not be appropriate under this situation since the variance of $\tilde{p}$ in this formula is only useful for Bernoulli trials when all parking cars have the same distribution and independently distributed. However, by picking blocks randomly, we may encounter differences between blocks (maybe some blocks have stricter rules for violation) and it will bring us the variance among all blocks.

(d) $\tilde{p}$ is not an unbiased estimator for p. For example, there are 3 blocks in a city and the number of cars

parked are 10, 8, 4 and the number of cars in violation of the meters are 10, 4, 0 respectively. We randomly select 2 blocks to estimate p. If we sample 1st and 2nd block, $\tilde{p} = 7/9$, 1st and 3rd: $\tilde{p} = 5/7$, 2nd and 3rd: $\tilde{p} = 1/3$. Then $\mathrm{E}(\tilde{p})$ is estimated by $(7/9 + 5/7 + 1/3)/3 = 115/189$, and p is estimated by $\frac{10+4}{10+8+4} = 7/11$. It's obvious that $\mathrm{E}(\tilde{p})$ is not equal to p.
Actually the unbiased estimator for p is $\hat{p} = \frac{N}{M}\frac{1}{n}\sum_{i=1}^{n} x_i$. Suppose N is the number of blocks in the population, n is the number of blocks selected, M is the number of cars ($\sum_{i=1}^{N} z_i = M$) and $\bar{M} = M/N$. Then an unbiased estimator for $\mathrm{Var}(\tilde{p})$ is $\frac{1}{M^2}N(N-n)\frac{1}{n}\hat{\sigma_b}^2$, where $\hat{\sigma_b}^2$ is the estimator for population variance of block totals and $\hat{\sigma_b}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(z_i\hat{p_i} - \bar{M}\hat{p})^2$.

Question3

```
samples <- matrix(c(4,12,3,8,3,9,3,16,2,7,4,15,4,10,3,15,2,6,1,12,3,8,2,14), ncol = 2, byrow = T)
df <- data.frame(samples)
#first we input the data as a object with x1=xi,x2=zi for next calculations.
```

(a)

```
alpha_hat <- t(apply(df, 2, sum)[1]/apply(df, 2, sum)[2]) # 0.2575758
# we sum the x1 and x2 separately and divide x1 by x2.
```

(b) Since in this problem, we do not know the specific sample distribution, and it is difficult to collect all the data at the same time.

(c) It samples the data from an empirical distribution, which is considered as the true distribution. The empirical distribution is a probability distribution that places a weight of 1/12 on each of the 12 data-points (x_i, z_i).

(d)

```
list1 <- c()
set.seed(1)#set seed in case to change the list every time
for (i in 1:1000){
  df1 <- df[sample(nrow(df), 12, replace = T), ]
  #Re extract 12 data for each list with 1000 times to form bootstrap samples
  p <- t(apply(df1, 2, sum)[1]/apply(df1, 2, sum)[2])
  #calculate each p-hat
  list1 <- c(list1, p)
  #save 1000 p-hat as a list
}
alpha_b <- mean(list1)
print(alpha_b)
```

```
## [1] 0.2579261
```

```
print(alpha_hat)
```

```
##             X1
## [1,] 0.2575758
```

The average of the $\hat{\alpha_i}^{b}$'s is close to $\hat{\alpha}$. Therefore the estimation procedure does not suffer an appreciable bias.

(e)

```
qu <- quantile(list1, probs = 0.975)
qd <- quantile(list1, probs = 0.025)
```

```
# lower bound
2*alpha_hat-qu
```

```
##             X1
## [1,] 0.2001702
```

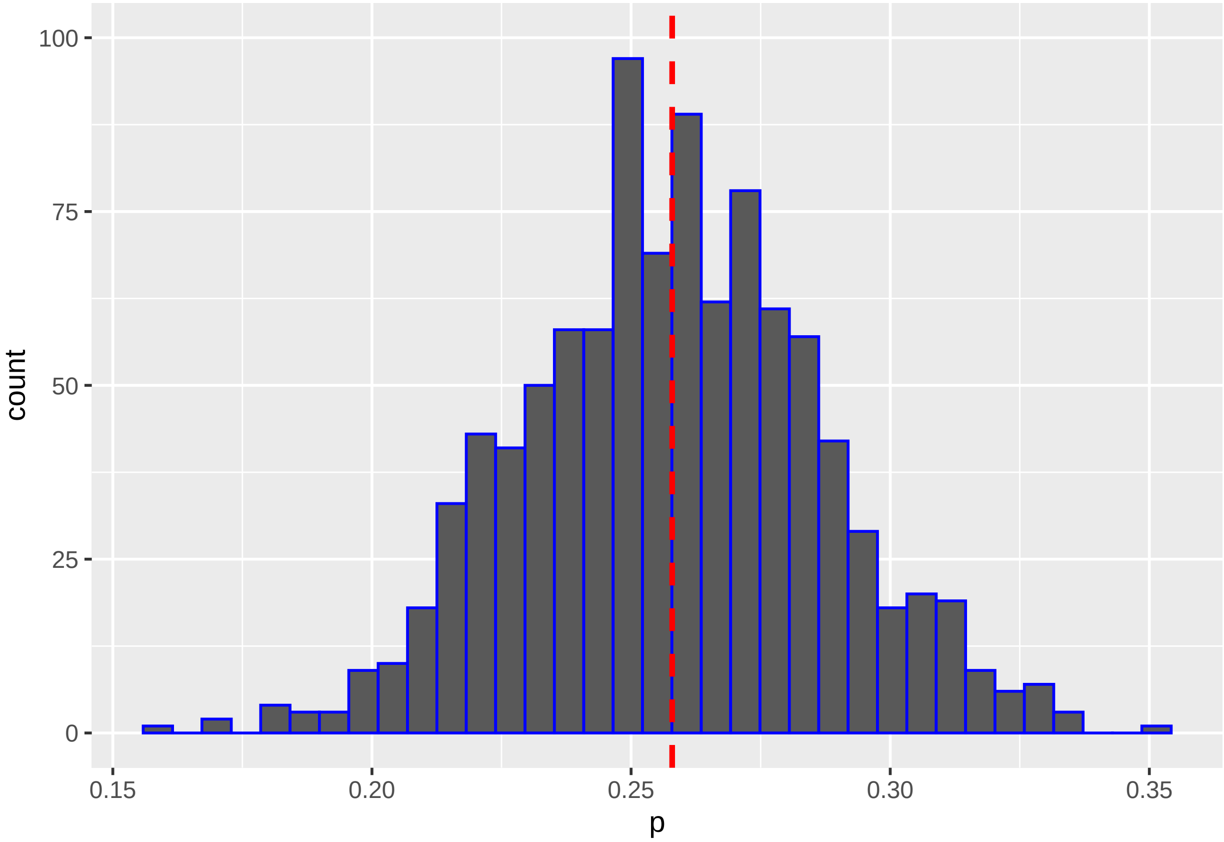
```
# upper bound
2*alpha_hat-qd
```

```
##             X1
## [1,] 0.3119056
```

(f)

```
df2 <- data.frame(list1)
library(ggplot2)

ggplot(df2, aes(x = list1)) +
  geom_histogram(colour="blue", bins = 35) + labs(x = 'p' , y = 'count') +
  scale_y_continuous(limits=c(0, 100)) +
  geom_vline(xintercept=alpha_b, color = "red", size = 1, linetype="dashed")
```



Because the histogram shows a symmetric distribution approximately, the data outside the confidence interval is a small probability event by default and will not occur.

Question 4

(b).

From the result of (a). we know that for optimization problem

$$\min_{\beta} E[q(Y-\beta)^{+} + (1-q)(Y-\beta)^{-}] \qquad (1)$$

The optimal solution (β^{*}) is the q-th quantile y_q of the random variable Y as any point that satisfies the equation $F(y_q) = q$, where $F(y)$ is the CDF of random variable Y.
According to the definition of CDF, $F(y) = Prob(Y \leq y)$.

For question (b), we are looking for the optimal solution of optimization problem

$$\min_{\{\beta(x):\, R^{d} \to R\}} E[q(Y-\beta(X))^{+} + (1-q)(Y-\beta)^{-}] \qquad (2)$$

As $\beta(x)$ is a function that map x from R^{d} to R, and $X \in R^{d}$, expression (1) and (2) are equivalent. While the q-th quantile y_q of the random variable Y is the optimal solution for (1), given condition $X = x$, the conditional q-quantile $\beta^{*}(x)$ of random variable Y is the optimal solution for expression (2).
Therefore, q-quantile $\beta^{*}(x)$ assess how much y will change for distribution y as x change by 1 unit at quantile point q for a given set of other covariates.

(c).

The goal is to find w that $\min_{w \in R^{d+1}} [\![y - Mw]\!]_1 + (2q-1)1^{T}(y - Mw)$

For $[\![y - Mw]\!]_1$, if $y > Mw, [\![y - Mw]\!]_1 = Y - Mw$
if $y < Mw, [\![y - Mw]\!]_1 = Mw - Y$

Therefore,

$$\min_{w \in R^{d+1}} [\![y - Mw]\!]_1 + (2q-1)1^{T}(y - Mw)$$

$$= min \sum_{i \,\epsilon\, y_i > M_i w_i}^{N} y_i - M_i w_i + (2q-1)(y_i - M_i w_i) + \sum_{i \,\epsilon\, y_i < M_i w_i}^{N} M_i w_i - y_i + (2q-1)(y_i - M_i w_i)$$

$$= min \sum_{i \,\epsilon\, y_i > M_i w_i}^{N} y_i - M_i w_i + (2q)(y_i - M_i w_i) - (1)(y_i - M_i w_i) + \sum_{i \,\epsilon\, y_i < M_i w_i}^{N} -(y_i - M_i w_i) + (2q)(y_i - M_i w_i) - (1)(y_i - M_i w_i)$$

$$= min \sum_{i \,\epsilon\, y_i > M_i w_i}^{N} (2q)(y_i - M_i w_i) + \sum_{i \,\epsilon\, y_i < M_i w_i}^{N} (2q-2)(y_i - M_i w_i)$$

According to what is defined in the question,

$\beta(x)$ is restricted to be of the form $\beta(x) = \left[X_i^{T} 1\right] * w_i$, and M_i is $[X_i^{T} 1]$

We could rewrite the expression

$$min \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} (2q)(y_i - M_i w_i) \, + \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} (2q-2)(y_i - M_i w_i)$$

$$= \; min \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} (2q)\big(y_i - \beta(x_i)\big) \, + \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} (2q-2)\big(y_i - \beta(x_i)\big)$$

$$= \; min \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} q\big(y_i - \beta(x_i)\big) \, + \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} (q-1)\big(y_i - \beta(x_i)\big)$$

$$= \; min \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} q\big(y_i - \beta(x_i)\big) \, + \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} (1-q)(-(y_i - \beta(x_i)))$$

Define $(x)^+ \;:= \max\{x, 0\}, when\ x < 0, (x)^+ = 0, and\ when\ x > 0, (x)^+ = x$

So

$$\sum_{i \, \epsilon \, y_i > M_i w_i}^{N} q\big(y_i - \beta(x_i)\big)$$

$$= \; q \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} 0 + \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} \big(y_i - \beta(x_i)\big)$$

$$= \; q \sum_{i}^{N} (y_i - \beta(x_i))^+$$

Define $(x)^- \;:= \max\{-x, 0\}, when\ x < 0, (x)^+ = -x, and\ when\ x > 0, (x)^+ = 0$

So

$$\sum_{i \, \epsilon \, y_i < M_i w_i}^{N} (1-q)(-(y_i - \beta(x_i)))$$

$$= (1-q) \sum_{i \, \epsilon \, y_i < M_i w_i}^{N} -\big(y_i - \beta(x_i)\big) + \sum_{i \, \epsilon \, y_i > M_i w_i}^{N} 0$$

$$= (1-q) \sum_{i}^{N} (y_i - \beta(x_i))^-$$

Therefore

$$
\begin{aligned}
&min \sum_{i \,\epsilon\, y_i > M_i w_i}^{N} q\big(y_i - \beta(x_i)\big) \; + \sum_{i \,\epsilon\, y_i < M_i w_i}^{N} (1-q)(-(y_i - \beta(x_i))) \\
&= \min q \sum_{i}^{N} (y_i - \beta(x_i))^{+} + (1-q) \sum_{i}^{N} (y_i - \beta(x_i))^{-} \\
&= \; \min \sum_{i}^{N} [q(y_i - \beta(x_i))^{+} + \; (1-q)(y_i - \beta(x_i))^{-} \;] \\
&= \; \min E[q(Y - \beta(X))^{+} + \; (1-q)(Y - \beta(X))^{-}]
\end{aligned}
$$

Therefore, we prove that $\min E[q(Y - \beta(X))^{+} + \; (1-q)(Y - \beta(X))^{-}]$

$$\approx \min_{w \,\epsilon\, R^{d+1}} [\![y - Mw]\!]_1 + (2q-1)1^{T}(y - Mw)$$