

Question1

$$(a)\widehat{\beta_1} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} \quad \widetilde{\beta_1} = \frac{\sum_{i=1}^n (\widetilde{X}_i - \bar{\widetilde{X}})(Y_i - \bar{Y})}{\sum_{i=1}^n (\widetilde{X}_i - \bar{\widetilde{X}})^2}$$

Based on the equation given by the question $\widetilde{X}_i = c(X_i + d)$, we can get the following equation.

$$\bar{\widetilde{X}} = c(\bar{X} + d).$$

Then focusing on the equation $\widetilde{\beta_1}$, we can get the result of $\widetilde{X}_i - \bar{\widetilde{X}}$.

$$\widetilde{X}_i - \bar{\widetilde{X}} = c(X_i + d) - c(\bar{X} + d) = c(X_i - \bar{X}).$$

Putting this result back to $\widetilde{\beta_1}$.

$$\begin{aligned} \widetilde{\beta_1} &= \frac{\sum_{i=1}^n c(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (c(X_i - \bar{X}))^2} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n c(X_i - \bar{X})^2} = \frac{1}{c} \widehat{\beta_1} \\ \widetilde{\beta_1} &= \frac{1}{c} \widehat{\beta_1} \end{aligned}$$

Using the same way, we can get the relationship between $\widetilde{\beta_0}$ and $\widehat{\beta_0}$

$$\begin{aligned} \widetilde{\beta_0} &= \bar{Y} - \widetilde{\beta_1} \bar{\widetilde{X}} \\ \widehat{\beta_0} &= \bar{Y} - \widehat{\beta_1} \bar{X} = \bar{Y} - \widehat{\beta_1} (\bar{X} + d) \\ \widetilde{\beta_0} - \widehat{\beta_0} &= \bar{Y} - \widetilde{\beta_1} \bar{\widetilde{X}} - \bar{Y} + \widehat{\beta_1} (\bar{X} + d) = \widehat{\beta_1} d \\ \widetilde{\beta_0} - \widehat{\beta_0} &= \widehat{\beta_1} d \end{aligned}$$

Also, we can get the relationship between \widetilde{Y}_i and \widehat{Y}_i .

$$\begin{aligned} \widehat{Y}_i &= \widehat{\beta_0} + \widehat{\beta_1} X_i \\ \widetilde{Y}_i &= \widetilde{\beta_0} + \widetilde{\beta_1} \widetilde{X}_i \end{aligned}$$

$$\widehat{Y}_i - \widetilde{Y}_i = \widehat{\beta_0} + \widehat{\beta_1} X_i - (\widetilde{\beta_0} + \widetilde{\beta_1} \widetilde{X}_i) = \widehat{\beta_1} d + \widehat{\beta_1} \left(X_i - \frac{1}{c} \times c(X_i + d) \right) = 0$$

Therefore, we can get the result that $\widehat{Y}_i = \widetilde{Y}_i$.

Using the result which is $\widetilde{y}_i = \widehat{y}_i$, we can get the relationship between $\widehat{\sigma}$ and $\widetilde{\sigma}$.

$$\begin{aligned} \widehat{\sigma} &= \sqrt{\frac{\sum (Y_i - \widehat{Y}_i)^2}{n - p}} \\ \widetilde{\sigma} &= \sqrt{\frac{\sum (Y_i - \widetilde{Y}_i)^2}{n - p}} \\ \widetilde{\sigma} &= \widehat{\sigma} \end{aligned}$$

Therefore, we can get the result that $\widetilde{\sigma} = \widehat{\sigma}$.

$$(b) \widehat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} \quad \widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{X}$$

From this Question, we can get there are 2 situations of $X_i = \begin{cases} 0 \\ 1 \end{cases}$. When $X_i = 0$, this is the situation \bar{Y}_p . When $X_i = 1$, this is the situation \bar{Y}_t .

Consider the situation T's number is n_t . Also, consider the situation P's number is n_p . Therefore, we can get the result of \bar{Y} .

$$\bar{Y} = \frac{n_t \cdot \bar{Y}_t + n_p \cdot \bar{Y}_p}{n_t + n_p}$$

$$\bar{X} = \frac{n_t \times 1 + n_p \times 0}{n_t + n_p} = \frac{n_t}{n_t + n_p}$$

Based on the $\widehat{\beta}_1$, we can calculate the result of $\sum_{i=1}^n (X_i - \bar{X})^2$ and $\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$.

$$\begin{aligned} \sum_{i=1}^n (X_i - \bar{X})^2 &= n_t \cdot (1 - \bar{X})^2 + n_p \cdot (0 - \bar{X})^2 \\ &= n_t \cdot \left(1 - \frac{n_t}{n_t + n_p}\right)^2 + n_p \cdot \left(0 - \frac{n_t}{n_t + n_p}\right)^2 \\ &= \frac{n_t n_p^2}{(n_t + n_p)^2} + \frac{n_p n_t^2}{(n_t + n_p)^2} = \frac{n_p n_t}{n_t + n_p} \\ \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) &= \sum_t (1 - \bar{X})(Y_i - \bar{Y}) - \sum_p \bar{X}(Y_i - \bar{Y}) \\ &= (1 - \bar{X}) \sum_t (Y_i - \bar{Y}) - \bar{X} \sum_p (Y_i - \bar{Y}) \\ &= (1 - \bar{X})(\bar{Y}_t - \bar{Y})n_t - \bar{X}n_p(\bar{Y}_p - \bar{Y}) \end{aligned}$$

Then, using simplify this equation we can get the result of $\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$.

$$\begin{aligned} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) &= n_t(\bar{Y}_t - \bar{Y}) - \bar{X}(n_t \bar{Y}_t + n_p \bar{Y}_p + \bar{Y}(n_t + n_p)) = n_t(\bar{Y}_t - \bar{Y}) \\ \widehat{\beta}_1 &= \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} = \frac{n_t(\bar{Y}_t - \bar{Y})}{\frac{n_p n_t}{n_t + n_p}} = \frac{(\bar{Y}_t - \bar{Y})(n_t + n_p)}{n_p} \\ &= \frac{\left(\bar{Y}_t - \frac{n_t \cdot \bar{Y}_t + n_p \cdot \bar{Y}_p}{n_t + n_p}\right)(n_t + n_p)}{n_p} \\ &= \frac{(\bar{Y}_t(n_t + n_p) - n_t \cdot \bar{Y}_t - n_p \cdot \bar{Y}_p)(n_t + n_p)}{(n_t + n_p)n_p} = \frac{np(\bar{Y}_t - \bar{Y}_p)(n_t + n_p)}{(n_t + n_p)n_p} \\ &= \bar{Y}_t - \bar{Y}_p \end{aligned}$$

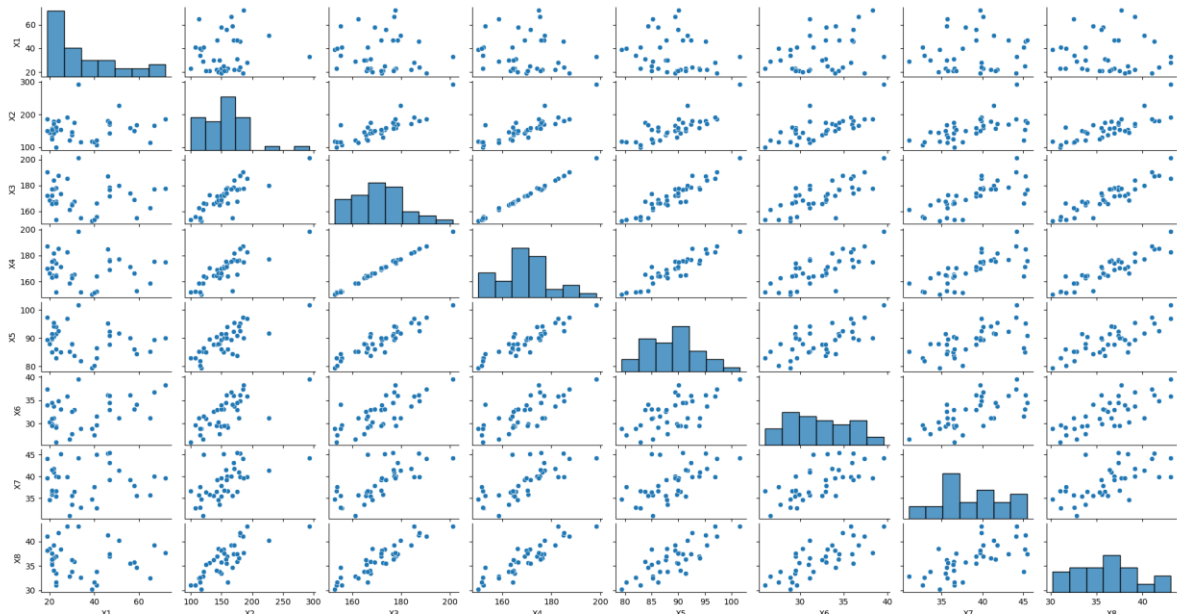
Therefore, $\widehat{\beta}_1 = \bar{Y}_t - \bar{Y}_p$

$$\widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{X} = \frac{n_t \cdot \bar{Y}_t + n_p \cdot \bar{Y}_p}{n_t + n_p} - \frac{n_t}{n_t + n_p} (\bar{Y}_t - \bar{Y}_p) = \bar{Y}_p$$

Therefore, $\widehat{\beta}_0 = \bar{Y}_p$

Question2

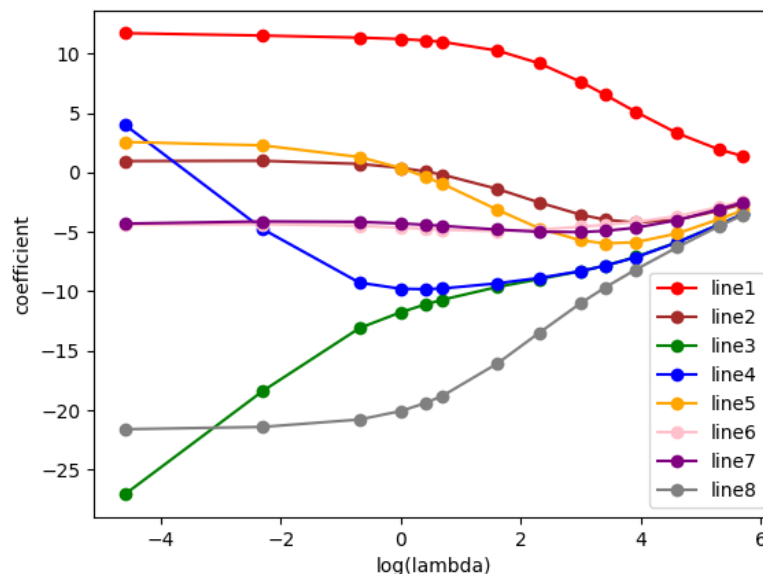
(a) From the figure given below, all the points are neatly distributed on the diagonal on the image of x_3 and x_4 . This means that x_3 and x_4 are highly correlated. Furthermore, x_3 and x_4 have data redundancy. Consider graphs which points are sparse and have uneven distribution, for example like x_1 and x_2 , this means the correlation between x_1 and x_2 is very low. This is the result we expect, which is good for us to do the following training and predicting.



(b) Here is the result of (b), you can also find these results by running the source code.

Question 2b result:
This is the result of checking(same with the number n) n: [38. 38. 38. 38. 38. 38. 38. 38.]

(c) Here is the result of (c). From the line chart, we can find that all the lines show a converging trend with the increase of $\log \lambda$. Based on question2 (a), we can find that line3 and line4 overlapped at around $\log \lambda = 2$. Line 5 is different from line3 and line4, it doesn't overlap with these 2 lines. But the 3 lines finally showed a converging trend.



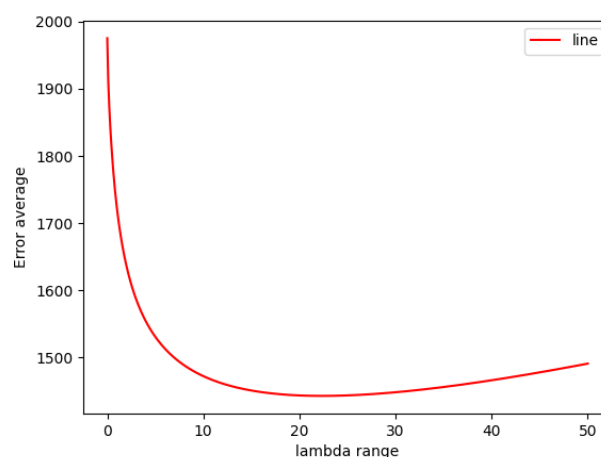
Here is the code screenshot of (c)

```

Question2.py X data.csv
hw01 > Question2.py > -
75 # This is the code for Question2 part c
76 # Using sklearn ridge to implement c
77 #####
78 from sklearn.linear_model import Ridge
79 alpha = [0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300]
80 X = rescaled_dataset
81 Y = np.array(Y)
82 result = list()
83 for item in alpha:
84     regression_q2 = Ridge(item)
85     regression_q2.fit(X,Y)
86     arr = regression_q2.coef_
87     result.append(arr.tolist())
88 #print(result)
89
90 purging_result = list()
91 for item in result:
92     for i in item:
93         purging_result.append(i)
94 #print(purging_result)
95
96 draw_y = list()
97 np_coefficient = np.array(purging_result)
98 # print(np_coefficient)
99
100 for i in range(col):
101     draw_y.append(np_coefficient[:,i])
102 draw_y = np.array(draw_y)
103 # print(draw_y)
104
105 log_lambda = [np.log(item) for item in alpha]
106 # print(log_lambda)
107
108 # Draw 8 lines to check the result
109 line1=plt.plot(log_lambda,draw_y[0],'red',marker='o',label='line1')
110 line2=plt.plot(log_lambda,draw_y[1],'brown',marker='o',label='line2')
111 line3=plt.plot(log_lambda,draw_y[2],'green',marker='o',label='line3')
112 line4=plt.plot(log_lambda,draw_y[3],'blue',marker='o',label='line4')
113 line5=plt.plot(log_lambda,draw_y[4],'orange',marker='o',label='line5')
114 line6=plt.plot(log_lambda,draw_y[5],'pink',marker='o',label='line6')
115 line7=plt.plot(log_lambda,draw_y[6],'purple',marker='o',label='line7')
116 line8=plt.plot(log_lambda,draw_y[7],'grey',marker='o',label='line8')
117 plt.xlabel('log(lambda)')
118 plt.ylabel('coefficient')
119 plt.legend()
120 plt.show()
121

```

(d) Here is the result of (d). By doing the calculation, we can find the maximum leave-one-out value is 1975.4147393421708 when $\lambda = 0$. Also, we can get the minimum leave-one-out value is 1442.6982227952926 when $\lambda = 22.3$. 1442.6982227952926 is greater than 1085.8364079, this means the leave-one-out average error is higher than standard OLS. This is due to the constraints we add during LOOCV. Compared with the standard OLS, the standard version has no constraints brought by the cross-validation which will have high accuracy and low error value.



Here is the code of LOOCV in part (d). You can also find this part of the code from the source code file.

```

150 #####
151 # This is the code for Question2 part d
152 #####
153 # print(row)
154 Error_list = list()
155 lambda_list = list()
156 for lambda in range(0,501,1):
157     #print(i/10)
158     current_lambda = lambda/10
159     lambda_list.append(current_lambda)
160     Error = 0
161     for pointer in range(0,row):
162         temp_X = np.copy(rescaled_dataset)
163         temp_Y = np.copy(Y)
164         Testing_X = temp_X[pointer]
165         Testing_Y = temp_Y[pointer]
166         Training_X = np.delete(temp_X,pointer,0)
167         Training_Y = np.delete(temp_Y,pointer,0)
168
169         reg_q2_d = Ridge(current_lambda)
170         reg_q2_d.fit(Training_X,Training_Y)
171         predict_y = reg_q2_d.predict([Testing_X])
172         error = np.square(predict_y-Testing_Y)
173         # print(error)
174         Error += error
175     Error_list.append(Error/row)
176
177 rescaled_Error_list = list()
178 for item in Error_list:
179     for itm in item:
180         for i in itm:
181             rescaled_Error_list.append(i)
182 # Checking the result of resizing the list
183 # print(rescaled_Error_list)
184
185 min_error_value = min(rescaled_Error_list)
186 max_error_value = max(rescaled_Error_list)
187 min_index = rescaled_Error_list.index(min_error_value)
188 max_index = rescaled_Error_list.index(max_error_value)
189 min_lambda = lambda_list[min_index]
190 max_lambda = lambda_list[max_index]
191
192 # printing the result of min and max error value
193 print(f'The minimum error value: {min_error_value}. The current lambda value : {min_lambda}')
194 print(f'The maximum error value: {max_error_value}. The current lambda value : {max_lambda}')
195
196 # These part of code is to printing the plot of question 2d
197 Question_2d_plot=plt.plot(lambda_list,rescaled_Error_list,'red',label='line')
198 plt.xlabel('lambda range')
199 plt.ylabel('Error average')
200 plt.legend()
201 plt.show()
202

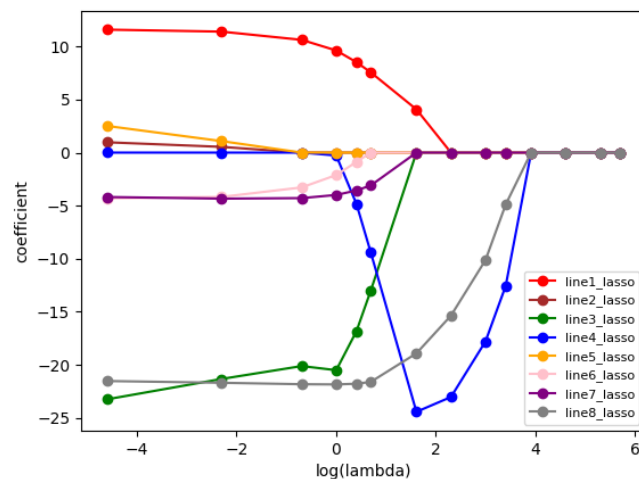
```

Here is the comparison result of (d)

Question 2d result:

The minimum error value of Ridge: 1442.6982227952915. The current lambda value of Ridge: 22.3
The maximum error value of Ridge: 1975.4147393421779. The current lambda value of Ridge: 0.0
standard linear regression: [1085.8364079]

(e) From this figure, we can find all these lines finally merge to the line directly. Line3 and line4 do not have a merging trend like question2 (c). In the same observation line4 and line5, when the $\log \lambda$ is close to 0, the two lines show a tendency to merge. Also, you can find that there is a surge in line4, when $0 < \log \lambda < 2$.



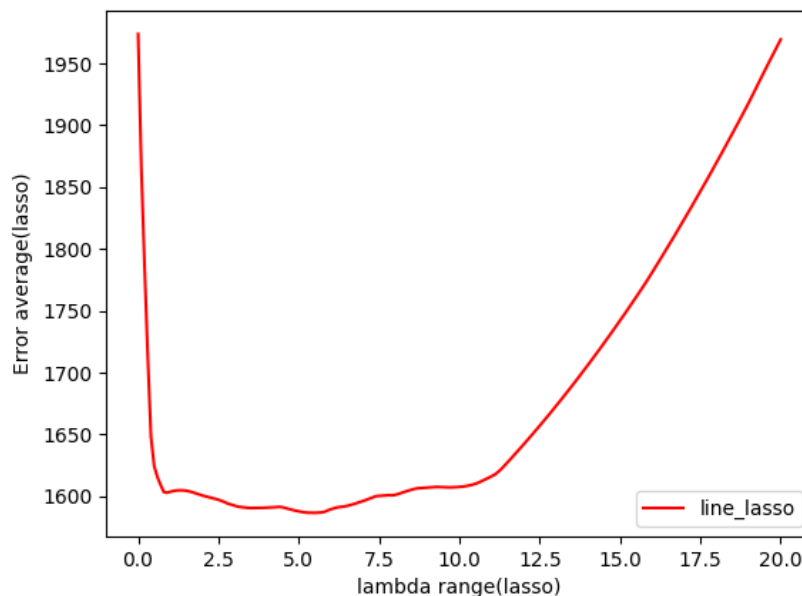
Here is the code of (e). You can also find this part of the code from the source code file. This part of the code is similar to the (c), the only difference is changing the Ridge to Lasso.

```

223 #####
224 # This is the code for Question2 part e
225 #####
226 from sklearn.linear_model import Lasso
227 alpha = [0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300]
228 result_lasso = list()
229 for item in alpha:
230     regression_lasso = Lasso(item)
231     regression_lasso.fit(X,Y)
232     lasso_coefficient = regression_lasso.coef_
233     result_lasso.append(lasso_coefficient.tolist())
234 #print(result_lasso)
235
236 np_coefficient_lasso = np.array(result_lasso)
237 # print(np_coefficient_lasso)
238
239 draw_y_lasso = list()
240 for i in range(col):
241     draw_y_lasso.append(np_coefficient_lasso[:,i])
242 np_draw_y_lasso = np.array(draw_y_lasso)
243 #print(np_draw_y_lasso)
244
245 log_lambda_lasso = [np.log(item) for item in alpha]
246
247 # Draw 8 lines to check the result
248 line1_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[0], 'red', marker='o', label='line1_lasso')
249 line2_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[1], 'brown', marker='o', label='line2_lasso')
250 line3_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[2], 'green', marker='o', label='line3_lasso')
251 line4_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[3], 'blue', marker='o', label='line4_lasso')
252 line5_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[4], 'orange', marker='o', label='line5_lasso')
253 line6_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[5], 'pink', marker='o', label='line6_lasso')
254 line7_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[6], 'purple', marker='o', label='line7_lasso')
255 line8_lasso=plt.plot(log_lambda_lasso,np_draw_y_lasso[7], 'grey', marker='o', label='line8_lasso')
256 plt.xlabel('log(lambda)')
257 plt.ylabel('coefficient')
258 plt.legend(fontsize=8)
259 plt.show()

```

(f) From this result, we can easily find the minimum leave-one-out error value of Lasso is greater than Ridge. Also, the maximum leave-one-out error value of Lasso is similar to Ridge. This means the performance of Lasso is worse than Ridge.



Here is the code of (f). You can also find this part of the code from the source code file. This part of the code is similar to the (d), the only difference is changing the Ridge to Lasso.

```

261 #####
262 # This is the code for Question2 part f
263 #####
264 Error_list_lasso = list()
265 lambda_list_lasso = list()
266 for lamda_lasso in range(0,201,1):
267     #print(l/10)
268     current_lambda_lasso = lamda_lasso/10
269     lambda_list_lasso.append(current_lambda_lasso)
270     Error_lasso = 0
271     for pointer_lasso in range(0,row):
272         temp_X_lasso = np.copy(rescaled_dataset)
273         temp_Y_lasso = np.copy(Y)
274         Testing_X_lasso = temp_X_lasso[pointer_lasso]
275         Testing_Y_lasso = temp_Y_lasso[pointer_lasso]
276         Training_X_lasso = np.delete(temp_X_lasso,pointer_lasso,0)
277         Training_Y_lasso = np.delete(temp_Y_lasso,pointer_lasso,0)
278
279         reg_q2f_lasso = Lasso(current_lambda_lasso)
280         reg_q2f_lasso.fit(Training_X_lasso,Training_Y_lasso)
281         predict_y_lasso = reg_q2f_lasso.predict([Testing_X_lasso])
282         error_lasso = np.square(predict_y_lasso-Testing_Y_lasso)
283         # print(error)
284         Error_lasso += error_lasso
285         Error_list_lasso.append(Error_lasso/row)
286 # print(Error_list_lasso)
287
288 purging_error_lasso = list()
289 for item in Error_list_lasso:
290     for i in item:
291         purging_error_lasso.append(i)
292 print(purging_error_lasso)
293 print(len(purging_error_lasso))
294
295 min_error_value_lasso = min(purging_error_lasso)
296 max_error_value_lasso = max(purging_error_lasso)
297 min_index_lasso = purging_error_lasso.index(min_error_value_lasso)
298 max_index_lasso = purging_error_lasso.index(max_error_value_lasso)
299 min_lambda_lasso = lambda_list_lasso[min_index_lasso]
300 max_lambda_lasso = lambda_list_lasso[max_index_lasso]
301
302 print("Question 2f result:")
303 # The minimum error value of Lasso: 1586.6715081806428. The current lambda value of Lasso : 5.5
304 # The maximum error value of Lasso: 1973.8286526002037. The current lambda value of Lasso : 0.0
305 print(f'The minimum error value of lasso: {min_error_value_lasso}. The current lambda value of lasso : {min_lambda_lasso}')
306 print(f'The maximum error value of lasso: {max_error_value_lasso}. The current lambda value of lasso : {max_lambda_lasso}')
307 print()
308
309 # These part of code is to printing the plot of question 2f
310 Question_2f_plot=plt.plot(lambda_list_lasso,purging_error_lasso,'red',label='line_lasso')
311 plt.xlabel('lambda range(lasso)')
312 plt.ylabel('Error average(lasso)')
313 plt.legend(loc='best')
314 plt.show()

```

Here is the error value of using Lasso in (e):

```

Question 2f result:
The minimum error value of lasso: 1586.6715081806424. The current lambda value of lasso : 5.5
The maximum error value of lasso: 1973.8286526002014. The current lambda value of lasso : 0.0

```

(g) From the graph of (c) and (e), the line graph of Ridge is smooth and gentle. Also, there is a fluctuation in the line graph of Lasso. From the graph of (d) and (f), the leave-one-out error value of Ridge is smooth and there is no clear upward trend. Furthermore, we can also find the fluctuation in graph (f). Also, at the same time, in the graph (f), we can find the line goes upward in Lasso. This is due to the difference in the penalty function. Ridge uses the L2 penalty which is to set the parameters as small as possible. However, Lasso uses the L1 penalty which is strict with the parameters which are forced the parameter to 0.

Question3

(a)

From the question, we can get the size of X is $n \times p$, the size of Y is $n \times 1$, the size of β is $p \times 1$. From the question, we can get the following equation.

$$|< Y, X\beta >| = |Y \cdot X\beta| = \left| \sum_i Y_i \cdot X_i \beta \right| = \left| \sum_i Y_i \cdot \sum_j X_{ij} \beta_j \right| = \left| \sum_i \sum_j Y_i \cdot X_{ij} \cdot \beta_j \right|$$

Then combining $Y_i \cdot X_{ij}$, we can get the following equation.

$$\left| \sum_i \sum_j Y_i \cdot X_{ij} \cdot \beta_j \right| = \left| \sum_j \sum_i (Y_i \cdot X_{ij}) \cdot \beta_j \right|$$

Due to $\sum_i Y_i \cdot X_{ij} = X_j^T Y$, we can simplify the equation given above.

$$\begin{aligned} \left| \sum_j \sum_i (Y_i \cdot X_{ij}) \cdot \beta_j \right| &= \left| \sum_j X_j^T Y \beta_j \right| = \sum_j |X_j^T Y| \times |\beta_j| \leq \sum_j |X_j^T Y| \times |\beta_j| \\ &\leq \sum_j \max_j |X_j^T Y| \times |\beta_j| \end{aligned}$$

In the equation given above, $\max_j |X_j^T Y|$ is a constant part. Therefore, we can finally get the result as showing below, which is the same as the result of this question.

$$|< Y, X\beta >| = \left| \sum_j \sum_i (Y_i \cdot X_{ij}) \cdot \beta_j \right| \leq \sum_j \max_j |X_j^T Y| \times |\beta_j| \leq \max_j |X_j^T Y| \sum_j |\beta_j|$$

(b)

$$\begin{aligned}\frac{1}{2}\|Y - \beta X\|_2^2 + \lambda\|\beta\|_1 &= \frac{1}{2}(Y - \beta X)^T(Y - \beta X) + \lambda\|\beta\|_1 \\&= \frac{1}{2}(Y^T - (\beta X)^T)(Y - \beta X) + \lambda\|\beta\|_1 \\&= \frac{1}{2}(Y^T Y + X\beta(X\beta)^T - 2Y^T \cdot X\beta) + \lambda\|\beta\|_1 \\&= \frac{1}{2}Y^T Y + \frac{1}{2}X\beta(X\beta)^T - Y^T \cdot X\beta + \lambda\|\beta\|_1 \\&= \frac{1}{2}Y^T Y + \frac{1}{2}X\beta(X\beta)^T - \langle Y, X\beta \rangle + \lambda \sum_j |\beta_j| \\&= \frac{1}{2}Y^T Y + \frac{1}{2} \left(X\beta(X\beta)^T - 2\langle Y, X\beta \rangle + 2\lambda \sum_j |\beta_j| \right) \\&= \frac{1}{2}Y^T Y + \frac{1}{2} \left(X\beta(X\beta)^T - 2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|) \right)\end{aligned}$$

Due to $\lambda \geq \max_j |X_j^T Y|$, also $|\langle Y, X\beta \rangle| \leq \max_j |X_j^T Y| \sum_j |\beta_j|$ and $\frac{1}{2}Y^T Y$ is the constant part of this equation, therefore, we can find out that $2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|) \leq 0$. Furthermore, $(X\beta(X\beta)^T - 2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|)) \geq 0$. To find out the minimum value of $\frac{1}{2}\|Y - \beta X\|_2^2 + \lambda\|\beta\|_1$, we can let $\frac{1}{2}(X\beta(X\beta)^T - 2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|)) = 0$, then the minimum value of this equation is $\frac{1}{2}Y^T Y$. To make $\frac{1}{2}(X\beta(X\beta)^T - 2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|)) = 0$, we can set $\hat{\beta} = 0_p$, then the value of $X\beta(X\beta)^T - 2(\langle Y, X\beta \rangle - \lambda \sum_j |\beta_j|)$ will become to 0, which can help $\frac{1}{2}\|Y - \beta X\|_2^2 + \lambda\|\beta\|_1$ to get the minimum value as $\frac{1}{2}Y^T Y$.

(c)

From the question, we can get this information $\beta \neq 0_p$, $\|X\beta\|_2 = 0$, $\|X\beta\|_2 > 0$. Using the information to prove $\ell(\beta) > \ell(0_p)$.

The first situation is $\beta \neq 0_p$, $\|X\beta\|_2 = 0$, then prove $\ell(\beta) > \ell(0_p)$.

$$\frac{1}{2}\|Y - \beta X\|_2^2 + \lambda\|\beta\|_1 = \frac{1}{2}Y^T Y + \frac{1}{2}X\beta(X\beta)^T - Y^T \cdot X\beta + \lambda\|\beta\|_1$$

If $\|X\beta\|_2 = 0$, then we can get $\frac{1}{2}X\beta(X\beta)^T = 0$, $Y^T \cdot X\beta = 0$.

However, $\lambda\|\beta\|_1 \geq \max_j |X_j^T Y| \|\beta\|_1 \neq 0$

Therefore, $\frac{1}{2}X\beta(X\beta)^T - Y^T \cdot X\beta + \lambda\|\beta\|_1 > 0$, from this equation, we can find out that $\frac{1}{2}Y^T Y + \frac{1}{2}X\beta(X\beta)^T - Y^T \cdot X\beta + \lambda\|\beta\|_1 > \frac{1}{2}Y^T Y$. This means we had already proved that $\ell(\beta) > \ell(0_p)$.

The second situation is $\beta \neq 0_p$, $\|X\beta\|_2 > 0$, then prove $\ell(\beta) > \ell(0_p)$.

$$\frac{1}{2}\|Y - \beta X\|_2^2 + \lambda\|\beta\|_1 = \frac{1}{2}Y^T Y + \frac{1}{2}(X\beta(X\beta)^T - 2(Y^T \cdot X\beta - \lambda\|\beta\|_1))$$

If $\|X\beta\|_2 > 0$, then we can get $-2(Y^T \cdot X\beta - \lambda\|\beta\|_1) \geq 0$ and $X\beta(X\beta)^T \geq 0$.

Therefore, $\frac{1}{2}(X\beta(X\beta)^T - 2(Y^T \cdot X\beta - \lambda\|\beta\|_1)) > 0$

$$\frac{1}{2}Y^T Y + \frac{1}{2}(X\beta(X\beta)^T - 2(Y^T \cdot X\beta - \lambda\|\beta\|_1)) > \frac{1}{2}Y^T Y$$

From the equation give above, when $\|X\beta\|_2 > 0$, we cannot get the minimum value as $\frac{1}{2}Y^T Y$.

This means when $\|X\beta\|_2 > 0$, we can prove that $\ell(\beta) > \ell(0_p)$.