Question1

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

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 $\widetilde{\beta_1}$.

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

$$\boldsymbol{\widetilde{\beta_1} = \frac{1}{c}\widehat{\beta_1}}$$

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

$$\widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X}$$

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

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

$$\boldsymbol{\widehat{\beta_0} - \widetilde{\beta_0} = \widehat{\beta_1}d}$$

Also, we can get the relationship between $\widetilde{Y_i}$ and $\widehat{Y_i}$.

$$\widehat{Y_i} = \widehat{\beta_0} + \widehat{\beta_1}X_i$$

$$\widetilde{Y_i} = \widetilde{\beta_0} + \widetilde{\beta_1}\widetilde{X_i}$$

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

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

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

$$\hat{\sigma} = \sqrt{\frac{\sum(Y_i - \widehat{Y_i})^2}{n - p}}$$

$$\tilde{\sigma} = \sqrt{\frac{\sum(Y_i - \widetilde{Y_i})^2}{n - p}}$$

$$\boldsymbol{\tilde{\sigma} = \hat{\sigma}}$$

Therefore, we can get the result that $\boldsymbol{\tilde{\sigma} = \hat{\sigma}}$.

(b) $\widehat{\beta_1} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sum_{i=1}^{n}(X_i-\bar{X})^2}$ $\widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X}$

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 $\overline{Y_p}$. When $X_i = 1$, this is the situation $\overline{Y_t}$.

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 $\bar{Y}$.

$$\boldsymbol{\bar{Y} = \frac{n_t \cdot \overline{Y_t} + n_p \cdot \overline{Y_p}}{n_t + n_p}}$$

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

Based on the $\widehat{\beta_1}$, we can calculate the result of $\sum_{i=1}^{n}(X_i - \bar{X})^2$and $\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$.

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

$$\begin{aligned}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) &= \sum_{t}(1-\bar{X})(Y_i - \bar{Y}) - \sum_{p}\bar{X}(Y_i - \bar{Y}) \\ &= (1-\bar{X})\sum_{t}(Y_i - \bar{Y}) - \bar{X}\sum_{p}(Y_i - \bar{Y}) \\ &= (1-\bar{X})(\overline{Y_t} - \bar{Y})n_t - \bar{X}n_p\left(\overline{Y_p} - \bar{Y}\right)\end{aligned}$$

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

$$\boldsymbol{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})} = n_t(\overline{Y_t} - \bar{Y}) - \bar{X}\left(n_t\overline{Y_t} + n_p\overline{Y_p} + \bar{Y}\left(n_t+n_p\right)\right) = \boldsymbol{n_t(\overline{Y_t} - \bar{Y})}$$

$$\begin{aligned}\widehat{\beta_1} &= \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} = \frac{n_t(\overline{Y_t} - \bar{Y})}{\frac{n_p n_t}{n_t+n_p}} = \frac{(\overline{Y_t} - \bar{Y})(n_t+n_p)}{n_p} \\ &= \frac{\left(\overline{Y_t} - \frac{n_t \cdot \overline{Y_t} + n_p \cdot \overline{Y_p}}{n_t+n_p}\right)(n_t+n_p)}{n_p} \\ &= \frac{\left(\overline{Y_t}\left(n_t+n_p\right) - n_t \cdot \overline{Y_t} - n_p \cdot \overline{Y_p}\right)(n_t+n_p)}{(n_t+n_p)n_p} = \frac{np(\overline{Y_t} - \overline{Y_p})(n_t+n_p)}{(n_t+n_p)n_p} \\ &= \overline{Y_t} - \overline{Y_p}\end{aligned}$$

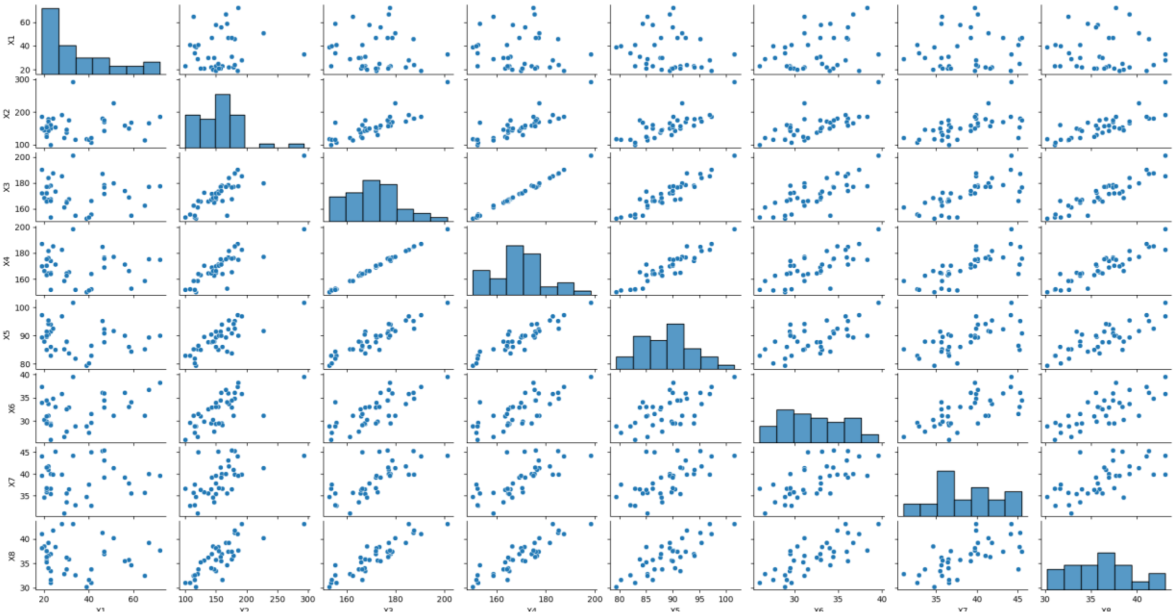
Therefore, $\boldsymbol{\widehat{\beta_1} = \overline{Y_t} - \overline{Y_p}}$

$$\widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X} = \frac{n_t \cdot \overline{Y_t} + n_p \cdot \overline{Y_p}}{n_t+n_p} - \frac{n_t}{n_t+n_p}\left(\overline{Y_t} - \overline{Y_p}\right) = \overline{Y_p}$$

Therefore, $\boldsymbol{\widehat{\beta_0} = \overline{Y_p}}$

Question2

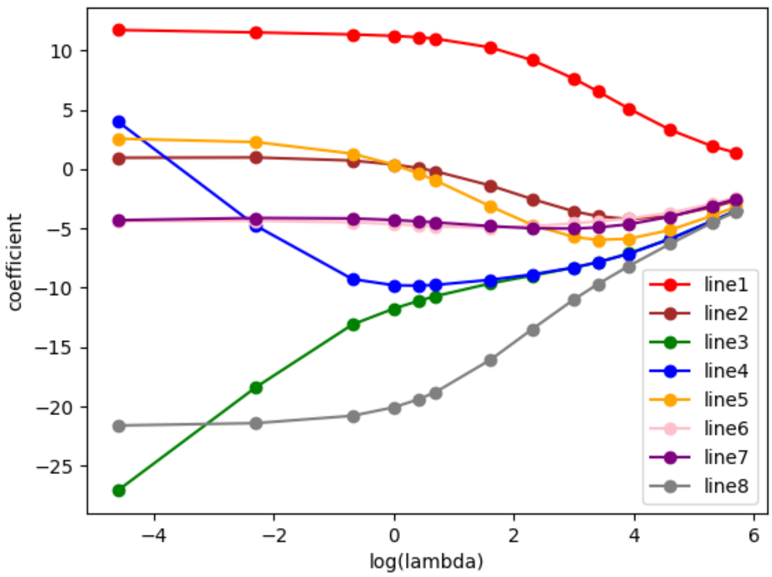
(a) From the figure given below, all the points are neatly distributed on the diagonal on the image of x3 and x4. This means that x3 and x4 are highly correlated. Furthermore, x3 and x4 have data redundancy. Consider graphs which points are sparse and have uneven distribution, for example like x1 and x2, this means the correlation between x1 and x2 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.]
```

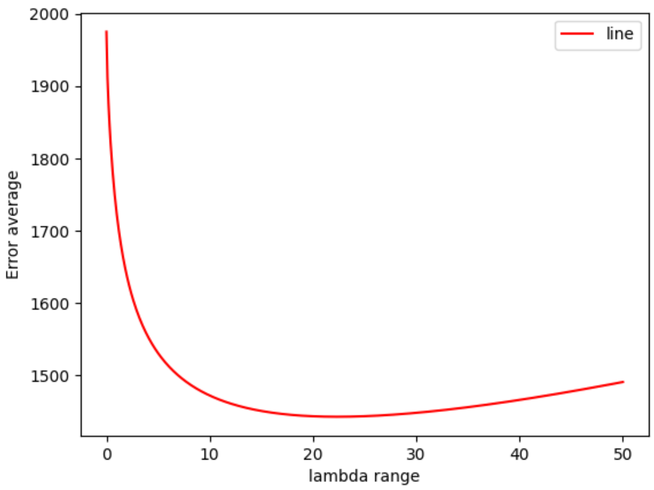
(c) Here is the result of (c). From the line chart, we can find that all the lines show a converging trend with the increase of $\log\lambda$. Based on question2 (a), we can find that line3 and line4 overlapped at around $\log\lambda = 2$. Line 5 is different from line3 and line4, it doesn't overlap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



Here is the code screenshot of (c)

```python
# This is the code for Question2 part c
# Using sklearn ridge to implement c
##########################################################################################
from sklearn.linear_model import Ridge
alpha = [0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300]
X = rescaled_dataset
Y = np.array(Y)
result = list()
for item in alpha:
    regression_q2 = Ridge(item)
    regression_q2.fit(X,Y)
    arr = regression_q2.coef_
    result.append(arr.tolist())
#print(result)

puring_result = list()
for item in result:
    for i in item:
        puring_result.append(i)
#print(puring_result)

draw_y = list()
np_coefficient = np.array(puring_result)
# print(np_coefficient)

for i in range(col):
    draw_y.append(np_coefficient[:,i])
draw_y = np.array(draw_y)
# print(draw_y)

log_lamda = [np.log(item) for item in alpha]
# print(log_lamda)

# Draw 8 lines to check the result
line1=plt.plot(log_lamda,draw_y[0],'red',marker='o',label='line1')
line2=plt.plot(log_lamda,draw_y[1],'brown',marker='o',label='line2')
line3=plt.plot(log_lamda,draw_y[2],'green',marker='o',label='line3')
line4=plt.plot(log_lamda,draw_y[3],'blue',marker='o',label='line4')
line5=plt.plot(log_lamda,draw_y[4],'orange',marker='o',label='line5')
line6=plt.plot(log_lamda,draw_y[5],'pink',marker='o',label='line6')
line7=plt.plot(log_lamda,draw_y[6],'purple',marker='o',label='line7')
line8=plt.plot(log_lamda,draw_y[7],'grey',marker='o',label='line8')
plt.xlabel('log(lambda)')
plt.ylabel('coefficient')
plt.legend()
plt.show()
```

(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.



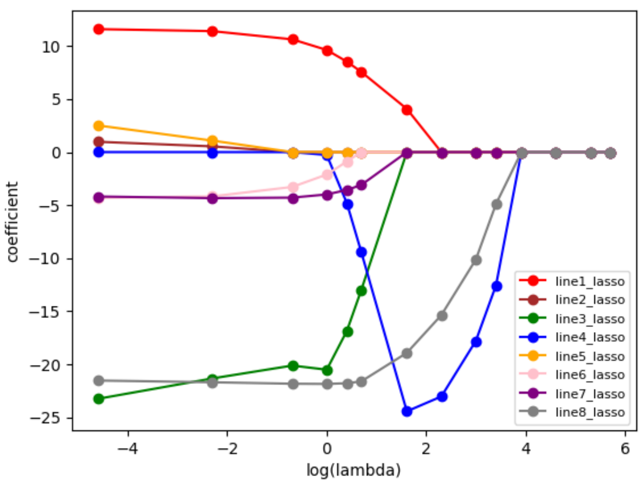
Here is the code of LOOCV in part (d). You can also find this part of the code from the source code file.

```python
##########################################################################################
# This is the code for Question2 part d
##########################################################################################
# print(row)
Error_list = list()
lambda_list = list()
for lamda in range(0,501,1):
    #print(i/10)
    current_lamda = lamda/10
    lambda_list.append(current_lamda)
    Error = 0
    for pointer in range(0,row):
        temp_X = np.copy(rescaled_dataset)
        temp_Y = np.copy(Y)
        Testing_X = temp_X[pointer]
        Testing_Y = temp_Y[pointer]
        Training_X = np.delete(temp_X,pointer,0)
        Training_Y = np.delete(temp_Y,pointer,0)


        reg_q2_d = Ridge(current_lamda)
        reg_q2_d.fit(Training_X,Training_Y)
        predict_y = reg_q2_d.predict([Testing_X])
        error = np.square(predict_y-Testing_Y)
        # print(error)
        Error += error
    Error_list.append(Error/row)

rescaled_Error_list = list()
for item in Error_list:
    for itm in item:
        for i in itm:
            rescaled_Error_list.append(i)
# Checking the result of resizing the list
# print(rescaled_Error_list)

min_error_value = min(rescaled_Error_list)
max_error_value = max(rescaled_Error_list)
min_index = rescaled_Error_list.index(min_error_value)
max_index = rescaled_Error_list.index(max_error_value)
min_lambda = lambda_list[min_index]
max_lambda = lambda_list[max_index]

# printing the result of min and max error value
print(f'The minimum error value: {min_error_value}. The current lambda value : {min_lambda}')
print(f'The maximum error value: {max_error_value}. The current lambda value : {max_lambda}')

# These part of code is to printing the plot of question 2d
Question_2d_plot=plt.plot(lambda_list,rescaled_Error_list,'red',label='line')
plt.xlabel('lambda range')
plt.ylabel('Error average')
plt.legend()
plt.show()
```

Here is the comparison result of (d)

```
Question 2d result:
The minimum error value: 1442.6982227952926. The current lambda value : 22.3
The maximum error value: 1975.4147393421708. The current lambda value : 0.0
standard linear regression: [1085.8364079]
```

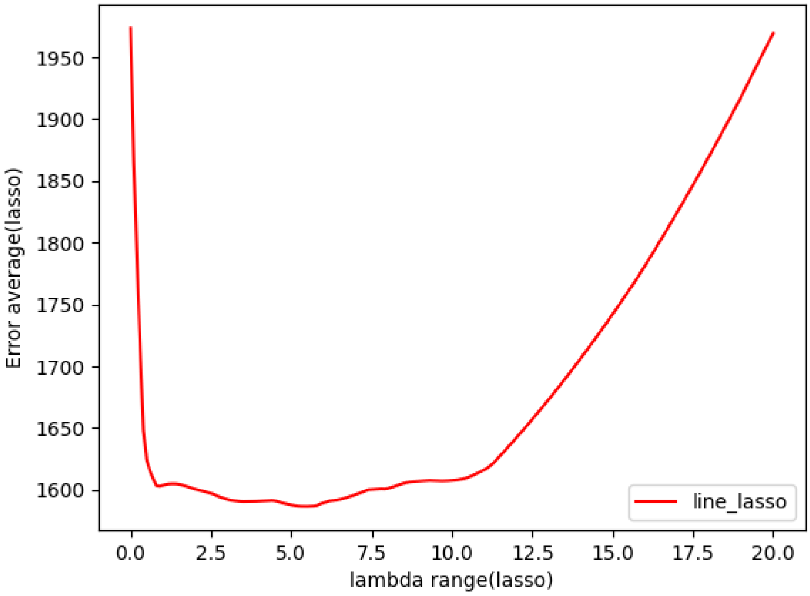
(e) From this figure, we can find all these lines finally merge to the line directly. Line3 and line4 do not have a merging trend like question2 (c). In the same observation line4 and line5, when the $\log\lambda$ is close to 0, the two lines show a tendency to merge. Also, you can find that there is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

```
######################################################################################
# This is the code for Question2 part e
######################################################################################
from sklearn.linear_model import Lasso
alpha = [0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300]
result_lasso = list()
for item in alpha:
    regression_lasso = Lasso(item)
    regression_lasso.fit(X,Y)
    lasso_coefficient = regression_lasso.coef_
    result_lasso.append(lasso_coefficient.tolist())
#print(result_lasso)

np_coefficient_lasso = np.array(result_lasso)
# print(np_coefficient_lasso)

draw_y_lasso = list()
for i in range(col):
   draw_y_lasso.append(np_coefficient_lasso[:,i])
np_draw_y_lasso = np.array(draw_y_lasso)
#print(np_draw_y_lasso)

log_lamda_lasso = [np.log(item) for item in alpha]

# Draw 8 lines to check the result
line1_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[0],'red',marker='o',label='line1_lasso')
line2_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[1],'brown',marker='o',label='line2_lasso')
line3_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[2],'green',marker='o',label='line3_lasso')
line4_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[3],'blue',marker='o',label='line4_lasso')
line5_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[4],'orange',marker='o',label='line5_lasso')
line6_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[5],'pink',marker='o',label='line6_lasso')
line7_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[6],'purple',marker='o',label='line7_lasso')
line8_lasso=plt.plot(log_lamda_lasso,np_draw_y_lasso[7],'grey',marker='o',label='line8_lasso')
plt.xlabel('log(lambda)')
plt.ylabel('coefficient')
plt.legend(fontsize=8)
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.

```python
##########################################################################################
# This is the code for Question2 part f
##########################################################################################
Error_list_lasso = list()
lambda_list_lasso = list()
for lamda_lasso in range(0,201,1):
    #print(i/10)
    current_lamda_lasso = lamda_lasso/10
    lambda_list_lasso.append(current_lamda_lasso)
    Error_lasso = 0
    for pointer_lasso in range(0,row):
        temp_X_lasso = np.copy(rescaled_dataset)
        temp_Y_lasso = np.copy(Y)
        Testing_X_lasso = temp_X_lasso[pointer_lasso]
        Testing_Y_lasso = temp_Y_lasso[pointer_lasso]
        Training_X_lasso = np.delete(temp_X_lasso,pointer_lasso,0)
        Training_Y_lasso = np.delete(temp_Y_lasso,pointer_lasso,0)

        reg_q2f_lasso = Lasso(current_lamda_lasso)
        reg_q2f_lasso.fit(Training_X_lasso,Training_Y_lasso)
        predict_y_lasso = reg_q2f_lasso.predict([Testing_X_lasso])
        error_lasso = np.square(predict_y_lasso-Testing_Y_lasso)
        # print(error)
        Error_lasso += error_lasso
    Error_list_lasso.append(Error_lasso/row)
# print(Error_list_lasso)

puring_error_lasso = list()
for item in Error_list_lasso:
    for i in item:
        puring_error_lasso.append(i)
print(puring_error_lasso)
print(len(puring_error_lasso))

min_error_value_lasso = min(puring_error_lasso)
max_error_value_lasso = max(puring_error_lasso)
min_index_lasso = puring_error_lasso.index(min_error_value_lasso)
max_index_lasso = puring_error_lasso.index(max_error_value_lasso)
min_lambda_lasso = lambda_list_lasso[min_index_lasso]
max_lambda_lasso = lambda_list_lasso[max_index_lasso]

print("Question 2f result:")
# The minimum error value of lasso: 1586.6715081806428. The current lambda value of lasso : 5.5
# The maximum error value of lasso: 1973.8286526002037. The current lambda value of lasso : 0.0
print(f'The minimum error value of lasso: {min_error_value_lasso}. The current lambda value of lasso : {min_lambda_lasso}')
print(f'The maximum error value of lasso: {max_error_value_lasso}. The current lambda value of lasso : {max_lambda_lasso}')
print()

# These part of code is to printing the plot of question 2f
Question_2f_plot=plt.plot(lambda_list_lasso,puring_error_lasso,'red',label='line_lasso')
plt.xlabel('lambda range(lasso)')
plt.ylabel('Error average(lasso)')
plt.legend(loc='best')
plt.show()
```

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

```
Question 2f result:
The minimum error value of lasso: 1586.6715081806428. The current lambda value of lasso : 5.5
The maximum error value of lasso: 1973.8286526002037. 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.

$$\left|\sum_j \sum_i (Y_i \cdot X_{ij}) \cdot \beta_j\right| = \left|\sum_j X_j^T Y \beta_j\right| = \sum_j \left|X_j^T Y\right| \times \left|\beta_j\right| \le \sum_j \left|X_j^T Y\right| \times \left|\beta_j\right|$$
$$\le \sum_j \max_j \left|X_j^T Y\right| \times \left|\beta_j\right|$$

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

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

(b)

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

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

(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 $\ell(\beta) > \ell(0_p)$.

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

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 $\frac{1}{2}Y^TY + \frac{1}{2}X\beta(X\beta)^T - Y^T \cdot X\beta + \lambda\|\beta\|_1 > \frac{1}{2}Y^TY$. 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 $\ell(\beta) > \ell(0_p)$.

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

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

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

From the equation give above, when $\|X\beta\|_2 > 0$, we cannot get the minimum value as $\frac{1}{2}Y^TY$. This means when $\|X\beta\|_2 > 0$, we can prove that $\ell(\beta) > \ell(0_p)$.