Q1:

(a):

$$\tilde{x}_i = c(x_i + d)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\tilde{\beta}_1 = \frac{\sum_{i=1}^{n}(\tilde{x}_i - \bar{\tilde{x}})(y_i - \bar{y})}{\sum_{i=1}^{n}(\tilde{x}_i - \bar{\tilde{x}})^2}$$

$$\because \bar{\tilde{x}} = c(\bar{x} + d) \qquad \tilde{x}_i = c(x_i + d)$$

$$\therefore \tilde{x}_i - \bar{\tilde{x}} = c(x_i + d) - c(\bar{x} + d) = c(x_i - \bar{x})$$

$$\therefore \tilde{\beta}_1 = \frac{\sum_{i=1}^{n}(\tilde{x}_i - \bar{\tilde{x}})(y_i - \bar{y})}{\sum_{i=1}^{n}(\tilde{x}_i - \bar{\tilde{x}})^2}$$

$$= \frac{c}{c^2} \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{c}{c^2} \cdot \hat{\beta}_1$$

$$= \frac{1}{c}\hat{\beta}_1 \qquad \therefore \tilde{\beta}_1 = \frac{1}{c}\hat{\beta}_1$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \qquad \tilde{\beta}_0 = \bar{y} - \tilde{\beta}_1 \bar{\tilde{x}}$$

$$\because \tilde{\beta}_1 = \frac{1}{c}\hat{\beta}_1 \qquad \bar{\tilde{x}} = c(\bar{x} + d)$$

$$\therefore \tilde{\beta}_0 = \bar{y} - \frac{1}{c}\hat{\beta}_1 \cdot c(\bar{x} + d)$$

$$\tilde{\beta}_0 = \bar{y} - (\bar{x} + d)\hat{\beta}_1 \qquad \tilde{\beta} = \hat{\beta}_0 - \hat{\beta}_1 d$$

$$\begin{cases} \hat{\sigma} = \sqrt{\dfrac{\sum(y_i - \hat{y}_i)^2}{n-p}} \\ \tilde{\sigma} = \sqrt{\dfrac{\sum(y_i - \tilde{y}_i)^2}{n-p}} \end{cases} \qquad \therefore \begin{cases} \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \\ \tilde{y}_i = \tilde{\beta}_0 + \tilde{\beta}_1 \tilde{x}_i \end{cases}$$

$$\tilde{\beta}_0 = (1 + \frac{d}{\bar{x}})\hat{\beta}_0 - \frac{d\bar{y}}{\bar{x}}$$

$$\tilde{\beta}_1 = \frac{1}{c}\hat{\beta}_1 \qquad \tilde{x}_i = c(x_i + d)$$

$$\tilde{y}_i = (1 + \frac{d}{\bar{x}})\hat{\beta}_0 - \frac{d\bar{y}}{\bar{x}} + (x_i + d)\hat{\beta}_1$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

$$\hat{\sigma} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-p}} \qquad \tilde{\sigma} = \sqrt{\frac{\sum(y_i - \tilde{y}_i)^2}{n-p}}$$

$$\therefore \hat{\sigma} = \sqrt{\frac{\sum(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{n-p}}$$

$$\tilde{\sigma} = \sqrt{\frac{\sum(y_i - \tilde{\beta}_0 - \tilde{\beta}_1 \tilde{x}_i)^2}{n-p}}$$

$$= \sqrt{\frac{\sum(y_i - (\frac{\bar{x}+d}{\bar{x}})\hat{\beta}_0 - \frac{\bar{y}}{\bar{x}}d + (x_i + d)\hat{\beta}_1)^2}{n-p}}$$

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

(b):

(b): $$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\bar{Y} = \frac{N_t \bar{Y}_t + N_p \cdot \bar{Y}_p}{N_t + N_p} = \frac{N_t}{N_t + N_p} \bar{Y}_t + \frac{N_p}{N_t + N_p} \bar{Y}_p$$

$$\bar{X} = \frac{N_t \cdot 1 + N_p \cdot 0}{N_t + N_p} = \frac{N_t}{N_t + N_p}$$

$$D = \sum_{i=1}^{n} (x_i - \bar{x})^2 = N_t (1 - \bar{x})^2 + N_p (0 - \bar{x})^2$$
$$= \frac{N_t N_p^2}{(N_t + N_p)^2} + \frac{N_p N_t^2}{(N_t + N_p)^2} = \frac{N_t \cdot N_p}{N_t + N_p}$$

$$M = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$
$$= \sum_t (1 - \bar{x})(y_j - \bar{y}) + \sum_p (0 - \bar{x})(y_i - \bar{y})$$
$$= N_t (1 - \bar{x})(\bar{Y}_t - \bar{y}) - \bar{x} N_p (\bar{Y}_p - \bar{y})$$

$$= N_t (\bar{Y}_t - \bar{y}) - \bar{x} \left[N_p (\bar{Y}_p - \bar{y}) + N_t (\bar{Y}_t - \bar{y}) \right]$$
$$= N_t (\bar{Y}_t - \bar{y})$$

$$\hat{\beta}_1 = N_t (\bar{Y}_t - \bar{y}) \cdot \frac{N_t + N_p}{N_t \cdot N_p} = \frac{N_t + N_p}{N_p} (\bar{Y}_t - \bar{y})$$

$$\because \bar{y} = \frac{N_t}{N_t + N_p} \bar{y}_t + \frac{N_p}{N_t + N_p} \bar{y}_p$$

$$\therefore \hat{\beta}_1 = \frac{N_t + N_p}{N_p} \left[\frac{N_p}{N_t + N_p} (\bar{y}_t - \bar{y}_p) \right]$$

$$\hat{\beta}_1 = \bar{y}_t - \bar{y}_p$$

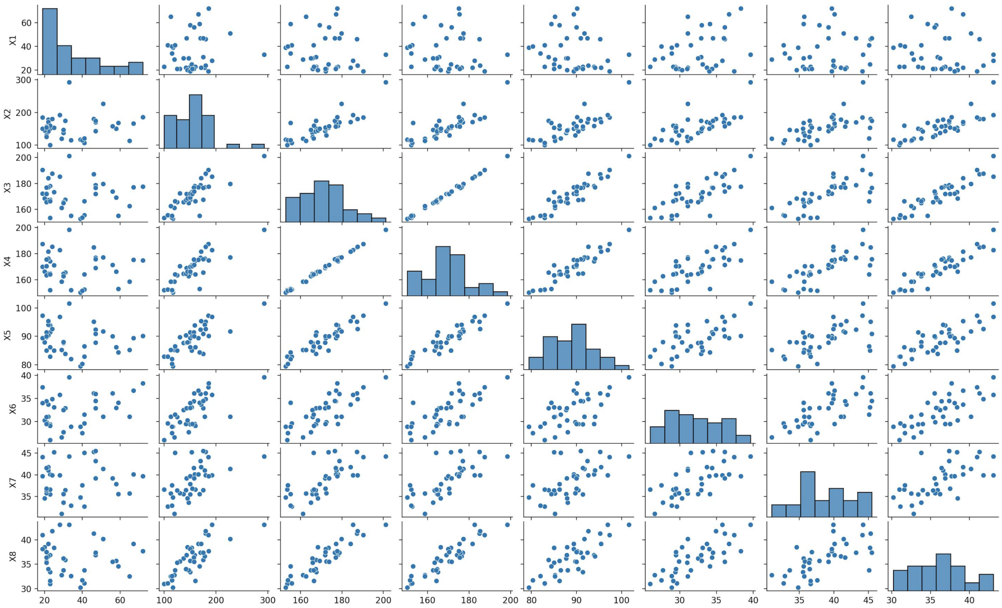
$$\because \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\therefore \hat{\beta}_0 = \bar{y} - \left(\frac{N_t}{N_t + N_p} \right) \cdot (\bar{y}_t - \bar{y}_p)$$

$$\hat{\beta}_0 = \frac{N_t}{N_t + N_p} \bar{Y}_t + \frac{N_p}{N_t + N_p} \bar{Y}_p - \frac{N_t}{N_t + N_p} (\bar{y}_t - \bar{y}_p)$$
$$= \frac{N_t}{N_t + N_p} \bar{Y}_p + \frac{N_p}{N_t + N_p} \bar{Y}_p = \bar{Y}_p$$

Q2:

(a):



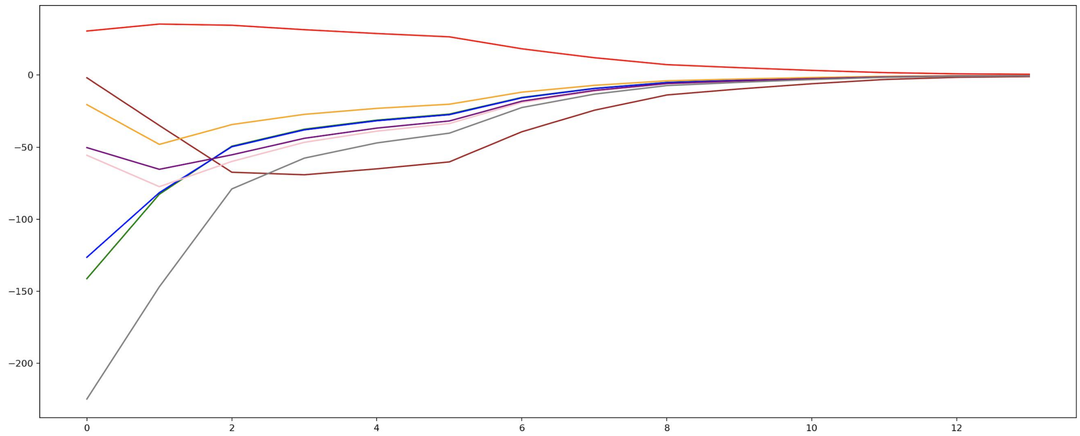
Some of the data are highly similar, which may affect the subsequent judgment

(b):

	X1	X2	X3	X4	X5	X6	X7	X8
0	0.279708	0.152697	0.092060	0.093340	0.070128	0.119931	0.171062	0.138305
1	-0.111061	0.121366	-0.022647	-0.021153	-0.057839	0.021126	-0.055485	-0.009972
2	-0.319471	-0.348599	-0.103582	-0.099645	-0.067942	-0.191922	-0.052911	-0.144519
3	-0.423676	0.184028	0.110110	0.108094	0.093700	0.160070	0.140170	0.130067
4	-0.319471	0.021107	0.038491	0.029602	0.055535	-0.083854	0.037193	0.017487
5	0.305760	0.090035	0.042567	0.046716	0.038697	0.116843	0.117000	0.031216
6	-0.137112	-0.116749	-0.033128	-0.026464	-0.014061	0.008775	-0.078655	-0.001734
7	-0.189214	0.227892	0.080997	0.080356	0.089210	0.110668	0.032045	0.187731
8	-0.319471	-0.035289	-0.022065	-0.024104	0.027472	-0.086942	-0.081229	-0.078619
9	-0.163163	-0.223275	-0.059330	-0.061284	-0.042124	-0.173396	-0.197078	-0.095094
10	0.305760	-0.079152	0.002973	0.000093	-0.012938	0.021126	0.014024	0.017487
11	0.149452	-0.304735	-0.091355	-0.097875	-0.067942	-0.080766	-0.153313	-0.141773
12	0.409965	0.447209	0.048972	0.047897	0.030840	-0.034452	0.070661	0.108100
13	-0.137112	-0.054087	-0.037786	-0.037678	-0.010693	-0.139432	-0.130143	-0.067635
14	-0.345522	0.140165	0.033833	0.043175	0.057780	-0.034452	0.060363	0.009249
15	0.826785	0.064971	0.033251	0.036684	0.005022	0.138457	0.037193	0.080642
16	-0.267368	-0.016490	0.011707	0.012487	-0.044369	0.027301	0.168488	0.058675
17	0.774682	-0.267138	-0.051178	-0.061284	-0.042124	-0.034452	-0.076080	-0.103331
18	-0.345522	-0.085418	-0.023812	-0.026464	0.016247	-0.083854	-0.055485	-0.062143
19	-0.371573	-0.160613	0.006466	0.008356	0.008389	-0.071503	-0.073506	-0.018209
20	-0.397625	-0.066620	-0.017407	-0.016432	-0.011816	-0.059153	-0.104399	0.061421
21	-0.058958	0.860778	0.173577	0.173012	0.141969	0.227998	0.142744	0.187731
22	-0.293420	0.152697	0.094389	0.095700	0.040942	0.082879	0.032045	0.152034
23	0.097349	-0.242073	-0.108241	-0.111449	-0.107230	-0.102380	-0.099250	-0.166486
24	0.592323	-0.035289	-0.012749	-0.015841	-0.030899	0.024214	-0.019444	-0.015464
25	-0.345522	0.096302	0.074009	0.073864	0.072373	0.045827	0.080958	0.080642
26	-0.371573	-0.191944	-0.032546	-0.033546	-0.044369	-0.037539	-0.058060	-0.026447
27	-0.319471	0.027373	-0.029052	-0.028235	-0.030899	-0.096205	-0.052911	-0.128044
28	-0.371573	0.008575	0.032668	0.037864	0.027472	0.067441	0.075809	0.003758
29	0.123401	-0.254606	-0.102418	-0.103186	-0.097127	-0.145607	-0.027167	-0.125298
30	0.618375	0.077503	-0.094266	-0.094924	-0.051104	0.058178	-0.078655	-0.045668
31	0.305760	0.121366	0.030339	0.039634	0.021860	0.070528	0.176211	0.031216
32	0.957041	0.190295	0.036744	0.034913	0.012879	0.187859	0.026896	0.039454
33	-0.032907	-0.254606	-0.094266	-0.099645	-0.078044	-0.102380	-0.148164	-0.100585
34	-0.423676	-0.035289	0.004719	0.004815	0.005022	0.055090	0.026896	0.050437
35	0.149452	-0.217008	-0.029634	-0.029415	-0.027531	-0.022101	0.165914	-0.067635
36	-0.371573	-0.010224	0.003555	0.007765	0.011757	-0.083854	-0.047762	0.033962
37	0.540221	0.014841	0.014036	0.014257	0.011757	0.119931	0.014024	-0.020955

$$\sum_i X_{ij}^2 \; for \; j = 1, \ldots, 8$$

(c):

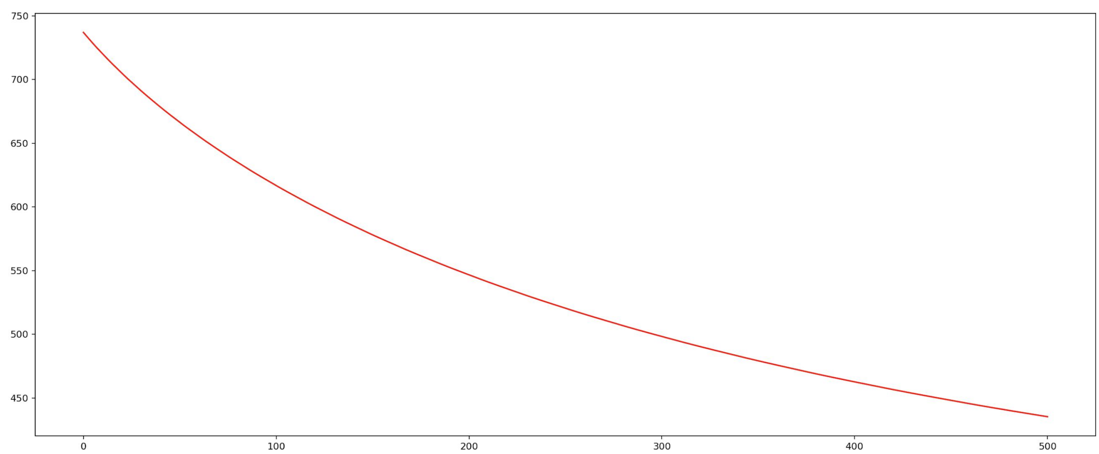



```python
lamda = [0.01,0.1,0.5,1,1.5,2,5,10,20,30,50,100,200,300]
all_lists = []
for i in lamda:
    Reg = Ridge(i)
    Reg.fit(new,y)
    all_lists.append(Reg.coef_)


matrix = np.array(all_lists)
```

```python
plt.figure(figsize=(20,8),dpi=80)

plt.plot(range(0,14),line0, color="red")
plt.plot(range(0,14),line1, color="brown")
plt.plot(range(0,14),line2, color="green")
plt.plot(range(0,14),line3, color="blue")
plt.plot(range(0,14),line4, color="orange")
plt.plot(range(0,14),line5, color="pink")
plt.plot(range(0,14),line6, color="purple")
plt.plot(range(0,14),line7, color="grey")

plt.show()
```

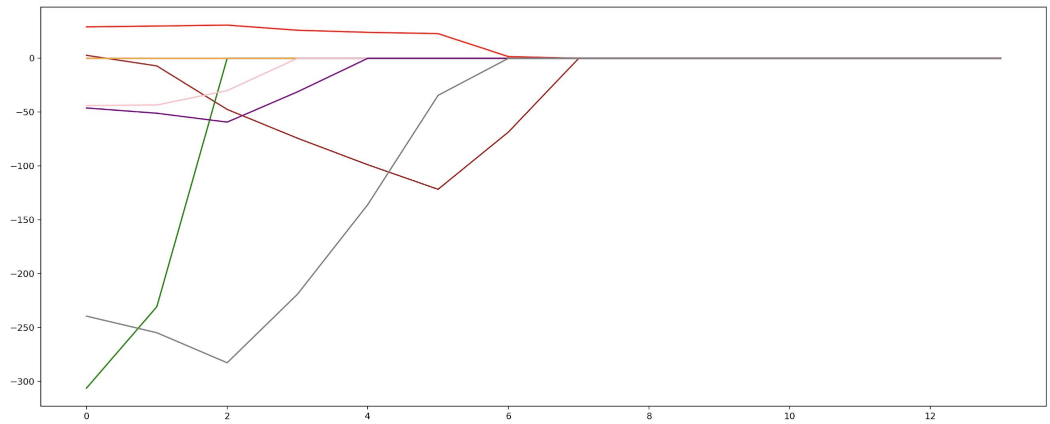
3,4,5The data is quickly collected, which means that the data is controlled quickly and the model works fast (d):

```
df = pd.read_csv("data.csv")
df = pd.DataFrame(df)
train_x=df.iloc[:-1, 0:-1]
train_y=df.iloc[:-1, -1:]
test_x=df.iloc[-1:, 0:-1]
test_y=df.iloc[-1:, -1:]

p=-0.1
E_list=[]
for i in range(0,501):
    p = p + 0.1
    alpha = round(p,1)
    E = 0
    for n in range(0,38):
        Reg = Ridge(alpha)
        Reg.fit(train_x,train_y)
        pre = Reg.predict(test_x)
        err = (pre - test_y)**2
        E = E + err
    E = E / 38
    E = E.values
    E_list.append(E[0][0])

plt.figure(figsize=(20,8),dpi=80)
plt.plot(range(0,501),E_list, color="red")
plt.show()
```



The data decreased slowly, indicating that the error value gradually narrowed and the data was close to the predicted value (e):

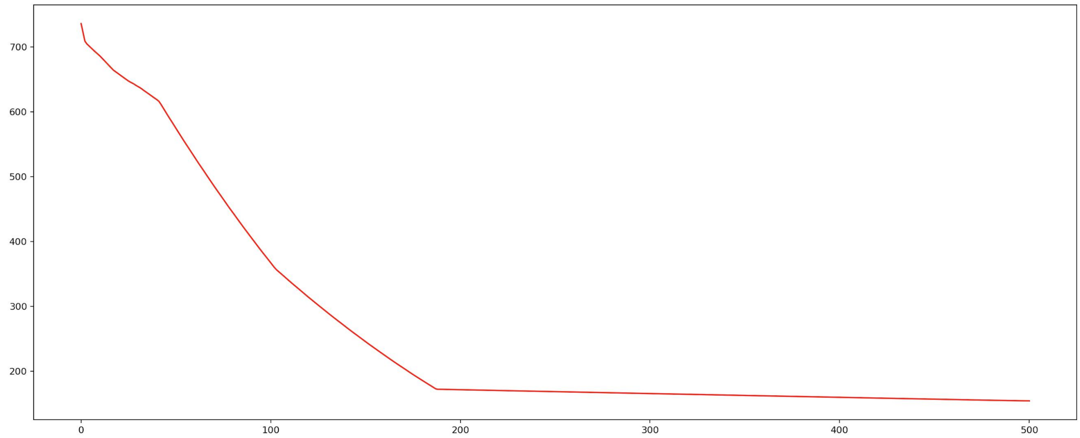
```
lamda = [0.01,0.1,0.5,1,1.5,2,5,10,20,30,50,100,200,300]
all_lists = []
for i in lamda:
    Reg = Lasso(i)
    Reg.fit(new,y)
    all_lists.append(Reg.coef_)

matrix = np.array(all_lists)
```



Compared with ridge, Lasso data are more divergent in the first half, but more consistent in the second half.Most of the data doesn't even change in the second half.

(f):

```
df = pd.read_csv("data.csv")
df = pd.DataFrame(df)
train_x=df.iloc[:-1, 0:-1]
train_y=df.iloc[:-1, -1:]
test_x=df.iloc[-1:, 0:-1]
test_y=df.iloc[-1:, -1:]

p=-0.1
E_list=[]
for i in range(0,501):
    p = p + 0.1
    alpha = round(p,1)
    E = 0
    for n in range(0,38):
        Reg = Lasso(alpha)
        Reg.fit(train_x,train_y)
        pre = Reg.predict(test_x)
        err = (pre - test_y)**2
        E = E + err
    E = E / 38
    E = E.values
    E_list.append(E[0][0])

print(E_list)
plt.figure(figsize=(20,8),dpi=80)
plt.plot(range(0,501),E_list, color="red")
plt.show()
```



The change of the first half is greater than that of the second half, and tends to be flat at last. Compared with ridge, Lasso has an obvious turning point, and the speed of data change is significantly different

(g):
Compared with lasso, ridge's curve is smoother, which is very obvious in the comparison of 2 (e) & 2 (c). Compared with messy Lasso, ridge's curve is more regular and easier to observe. But in both ridge and Lasso, the end of the data is highly consistent

Q3:

(a):

$$|\langle Y, X\beta\rangle| = |\sum_i Y_i \cdot X_i \beta|$$

$$= |\sum_i Y_i \sum_j X_{ij} \beta_j|$$

$$= |\sum_j [\sum_i (Y_i X_{ij})] \cdot \beta_j|$$

$$\because \sum_i X_{ij} Y_i = X_j^T Y$$

$$\therefore |X_j^T Y \cdot \sum_j \beta_j| = |\langle Y, X\beta\rangle|$$

$$\because X_j^T Y \le \max |X_j^T Y|$$

$$\therefore |\langle Y, X\beta\rangle| \le \max_j |X_j^T Y| \sum_j |\beta_j|$$

(b):

$$(b) \quad \frac{1}{2}(Y - X\beta)^2 = \frac{1}{2}(Y - X\beta)^T (Y - X\beta)$$

$$= \frac{1}{2}(Y^T - (X\beta)^T)(Y - X\beta)$$

$$= \frac{1}{2}(Y^T Y + (X\beta)^T) X\beta - Y^T X\beta$$

$$\because |\langle Y, X\beta\rangle| \le \max_j |X_j^T Y| \sum_j |\beta_j|$$

$$\because Y^T X\beta + \lambda \|\beta\|_1$$

$$(\langle Y, X\beta\rangle - \lambda \sum_j |\beta_j|)$$

$$\because \lambda \ge \max_j |X^T Y|$$

$$\therefore \lambda \ge X_j^T Y$$

$$\therefore \langle Y, X\beta\rangle - \lambda \sum_j |\beta_j| \ge 0$$

(c):

(c), $\because |X\beta_1| \geq 0$

if $\|X\beta_1\|_2 = 0$ $\quad \because \beta \neq 0_p$

$\hat{\beta} = \min \{ \frac{1}{2} \| Y - X\beta \|_2^2 + \lambda \|\beta\|_1$

if $\|X\beta_1\|_2 = 0$

$\therefore \frac{1}{2}(X\beta)^T X\beta = 0 \quad -Y^T X\beta = 0$

$\because \lambda \neq 0_p \quad \therefore \lambda \|\beta\|_1 > 0$

$\therefore \hat{\beta} > 0_p$

if $\| X\beta_1 \| > 0$

$\therefore \frac{1}{2}(X\beta)^T X\beta > 0 \quad \frac{1}{2} Y Y^T - Y^T X\beta > 0$

$\lambda \|\beta_1\| > 0$

$\therefore \hat{\beta} > 0_p$ $\quad \boxed{\therefore \text{ if } \beta \neq 0_p \text{ then } L(\beta) > L(0_p)}$