

COMP9417 – Machine Learning and Data Mining  
T2–2021

Homework 1: Linear Regression & Friends

Daksh Mukhra  
Z5163002

Q1)

a) We shall derive  $\tilde{\beta}_0, \tilde{\beta}_1$  from the Least Squares minimization criterion.

→ note we have univariate model.

$$L(\tilde{\beta}_0, \tilde{\beta}_1) = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 \tilde{x}_i)^2$$

$$= \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))^2 \quad // \text{as } \tilde{x} = c(x_i + d)$$

$$= \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))^2$$

→ differentiate wrt  $\tilde{\beta}_0$

$$\frac{\partial L}{\partial \tilde{\beta}_0} = \frac{\partial}{\partial \tilde{\beta}_0} \cdot \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))^2$$

$$= \frac{1}{n} \sum_{i=1}^n -2(y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))$$

$$= -2 \left[ \frac{1}{n} \sum_{i=1}^n y_i - \frac{\tilde{\beta}_0}{n} \sum_{i=1}^n 1 - \frac{\tilde{\beta}_1 c}{n} \sum_{i=1}^n (x_i + d) \right]$$

$$= -2 \left[ \bar{y} - \tilde{\beta}_0 - \tilde{\beta}_1 c \bar{x} - \tilde{\beta}_1 c d \right]$$

→ set = 0 & solve.

$$\bar{y} - \tilde{\beta}_0 - \tilde{\beta}_1 c \bar{x} - \tilde{\beta}_1 c d = 0$$

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

→ differentiate wrt  $\tilde{\beta}_1$

$$\frac{\partial L}{\partial \tilde{\beta}_1} = \frac{\partial}{\partial \tilde{\beta}_1} \cdot \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))^2$$

$$= -2 \frac{1}{n} \sum_{i=1}^n -2c(x_i + d)(y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))$$

$$= -2 \left[ \frac{1}{n} \sum_{i=1}^n c(x_i + d)y_i - \tilde{\beta}_0 c(x_i + d) - \tilde{\beta}_1 c^2 (x_i + d)^2 \right]$$

$$= -2 \left[ \frac{1}{n} \sum_{i=1}^n c x_i y_i + c d y_i - \tilde{\beta}_0 c x_i - \tilde{\beta}_0 c d - \tilde{\beta}_1 c^2 x_i^2 - \tilde{\beta}_1 c^2 d^2 - 2 \tilde{\beta}_1 c^2 x_i d \right]$$

$$\frac{\partial L}{\partial \tilde{\beta}_1} = -2 \left[ \frac{1}{n} \sum_{i=1}^n c x_i y_i + c d y_i - \tilde{\beta}_0 c d - \tilde{\beta}_0 c x_i - \tilde{\beta}_1 c^2 x_i^2 - \tilde{\beta}_1 c^2 d^2 - 2 \tilde{\beta}_1 c^2 x_i d \right]$$

$$\frac{\partial L}{\partial \tilde{\beta}_1} = -2 \left[ c \bar{xy} + c d \bar{y} - \tilde{\beta}_0 c d - \tilde{\beta}_0 c \bar{x} - \tilde{\beta}_1 c^2 \bar{x}^2 - \tilde{\beta}_1 c^2 d^2 - 2 \tilde{\beta}_1 c^2 d \bar{x} \right]$$

→ equate to 0 & solve

$$c d \bar{y} + c \bar{xy} - \tilde{\beta}_0 c d - \tilde{\beta}_0 c \bar{x} - \tilde{\beta}_1 c^2 \bar{x}^2 - \tilde{\beta}_1 c^2 d^2 - 2 \tilde{\beta}_1 c^2 d \bar{x} = 0$$

$$c d \bar{y} + c \bar{xy} - \tilde{\beta}_0 c d - \tilde{\beta}_0 c \bar{x} - \tilde{\beta}_1 c^2 \bar{x}^2 - \tilde{\beta}_1 c^2 d^2 - 2 \tilde{\beta}_1 c^2 d \bar{x} = 0$$

$$\tilde{\beta}_1 = \left[ \frac{\tilde{\beta}_0 c (d + \bar{x}) - c (d \bar{y} + \bar{xy})}{c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x}} \right] \times -1$$

$$\text{Sub in } \tilde{\beta}_0 = \bar{y} - \tilde{\beta}_1 c (\bar{x} + d)$$

$$\tilde{\beta}_1 = \frac{c (d \bar{y} + \bar{xy}) - [\bar{y} - \tilde{\beta}_1 c (\bar{x} + d)] c (d + \bar{x})}{c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x}}$$

$$\tilde{\beta}_1 = \frac{c (d \bar{y} + \bar{xy}) - c (d + \bar{x}) \bar{y} + \tilde{\beta}_1 c^2 (\bar{x} + d)^2}{c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x}}$$

$$\tilde{\beta}_1 [c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x}] = c (d \bar{y} + \bar{xy}) - c \bar{y} (d + \bar{x}) + \tilde{\beta}_1 c^2 (\bar{x} + d)^2$$

$$\tilde{\beta}_1 [c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x}] - \tilde{\beta}_1 c^2 (\bar{x} + d)^2 = c (d \bar{y} + \bar{xy}) - c \bar{y} (d + \bar{x})$$

$$\tilde{\beta}_1 c^2 [c^2 \bar{x}^2 + c^2 d^2 + 2 c^2 d \bar{x} - \bar{x}^2 - d^2 - 2 d \bar{x} d] = c (d \bar{y} + \bar{xy}) - c \bar{y} (d + \bar{x})$$

$$\tilde{\beta}_1 c^2 [\bar{x}^2 - \bar{x}^2] = c (d \bar{y} + \bar{xy}) - c \bar{y} (d + \bar{x})$$

$$\tilde{\beta}_1 = \frac{c \bar{y} + \bar{xy} - \bar{y} d - \bar{y} \bar{x}}{c [\bar{x}^2 - \bar{x}^2]} = \frac{\bar{xy} - \bar{xy}}{c [\bar{x}^2 - \bar{x}^2]}$$

$$\therefore \boxed{\tilde{\beta}_2 = \frac{1}{c} [\hat{\beta}_1]} \quad \text{where } \hat{\beta}_1 \text{ is the OLS estimator of } \beta_1 \text{ for } Y \sim X.$$

From earlier

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

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

$$\boxed{\tilde{\beta}_0 = \hat{\beta}_0 + d} \quad \text{where } \hat{\beta}_0 \text{ is the OLS estimator of } \beta_0 \text{ for } Y \sim X.$$

$\rightarrow$  we now find error variance:

$$\tilde{\sigma} = \sqrt{\frac{\tilde{e}^T \tilde{e}}{n-2}} \quad ; \quad \hat{\sigma} = \sqrt{\frac{\hat{e}^T \hat{e}}{n-2}}$$

$$\tilde{\sigma}^2 = \frac{\tilde{e}^T \tilde{e}}{n-2} = \frac{(y_i - \tilde{y}_i)^T (y_i - \tilde{y}_i)}{n-2}$$

$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{n-2} = \frac{\sum_{i=1}^n (y_i - (\tilde{\beta}_0 + \tilde{\beta}_1 x_i))^2}{n-2}$$

$$\rightarrow \frac{\sum_{i=1}^n (y_i - (\tilde{\beta}_0 + \tilde{\beta}_1 (c(x_i + d))))^2}{n-2} = \tilde{\sigma}^2 \quad // \text{sub in } \tilde{\beta}_0, \tilde{\beta}_1$$

$$\rightarrow \frac{\sum_{i=1}^n (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 c(x_i + d))^2}{n-2} = \frac{\sum_{i=1}^n (y_i - (\hat{\beta}_0 + d) - [\frac{1}{c} \hat{\beta}_1] c(x_i + d))^2}{n-2}$$

$$\rightarrow \frac{\sum_{i=1}^n (y_i - \hat{\beta}_0 - d - \hat{\beta}_1 x_i - \hat{\beta}_1 d)^2}{n-2} = \frac{\sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) - d(\hat{\beta}_1 + 1))^2}{n-2}$$

$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^n (y_i - \bar{y} - d(\hat{\beta}_1 + 1))^2}{n-2} = \boxed{\frac{\sum_{i=1}^n (\hat{e}_i - d(\hat{\beta}_1 + 1))^2}{n-2}}$$

Q(1) b)

→ let  $x_i = \begin{cases} 1 & \text{dosage if } i = n_1 + 1, \dots, n \\ 0 & \text{no dosage if } i = 1, \dots, n_1 \end{cases}$

→  $\bar{Y}_T$  is the sample mean of treatment group

$\bar{Y}_P$  is the mean of placebo group.

→ let  $n_2 + n_1 = n \therefore n_2 = (n - n_1)$

Hence let  $n_2$  be the number of patients that took dosage.  
 $n_1$  be the number of patients that took placebo.

→ we have the regression model  $y = \beta_0 + \beta_1 x + \epsilon$  where  $\epsilon \sim N(0, 1)$

→ From the usual minimization procedure we obtain the OLS estimates:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta} \bar{x} \quad \text{and} \quad \hat{\beta}_1 = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x}^2 - \bar{x}^2}$$

→ Note that the group means are.

$$\bar{Y}_P = \frac{\sum_{i=1}^{n_1} y_i}{n_1} \quad \bar{Y}_T = \frac{\sum_{i=n_1+1}^n y_i}{n_2}$$

→ Also Note the following derivations:

$$\textcircled{1} \quad - \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{n} \cdot n_2 = \frac{n_2}{n_1 + n_2} \quad // \text{only people who are treated} = 1 \quad \text{as}$$

$$\textcircled{2} \quad - (\bar{x})^2 = \left( \frac{n_2}{n_1 + n_2} \right)^2$$

$$\textcircled{3} \quad - \sum_{i=1}^n x_i^2 = \sum_{i=1}^n 1^2 + 0 + 0^2 + \dots = n_2.$$

$$\textcircled{4} \quad - \bar{x}\bar{y} = \frac{n_2}{n_1 + n_2} \left( \frac{1}{n} \sum_{i=1}^n y_i \right)$$

$$\textcircled{5} \quad - \sum_{i=1}^n x_i y_i = \sum_{i=1}^{n_1} (0) y_i + \sum_{i=n_1+1}^n (1) y_i = \sum_{i=1}^{n_2} y_i \quad // \text{as } n_2 = (n - n_1)$$

→ [see next pg.]

→ we now find  $\hat{\beta}_1$  in terms of group mean.

$$\rightarrow \hat{\beta}_{1, \text{OLS}} = \frac{1}{n} \sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i / n^2}{\frac{1}{n} \sum_{i=1}^n x_i^2 - \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2}$$

// where  $n = n_1 + n_2$   
using ①, ③  
④, ⑤

$$\hat{\beta}_1 = \frac{\frac{1}{n_1+n_2} \sum_{i=1}^{n_2} y_i - \left( \frac{n_2}{n_1+n_2} \right) \sum_{i=1}^{n_1} y_i \cdot \left( \frac{1}{n_1+n_2} \right)}{\frac{n_2}{n_1+n_2} - \left( \frac{n_2}{n_1+n_2} \right)^2}$$

from ②  
from ③

→ as  $n = n_1 + n_2$

$$\hat{\beta}_1 = \frac{\frac{1}{n_1+n_2} \sum_{i=1}^{n_2} y_i - \frac{n_2}{n_1+n_2} \left( \frac{1}{n_1+n_2} \cdot \left( \sum_{i=1}^{n_1} y_i + \sum_{i=1}^{n_2} y_2 \right) \right)}{n_2(n_1+n_2) - n_2^2}$$

$$\hat{\beta}_1 = \frac{\left( \frac{1}{n_1+n_2} \right) \left[ \sum_{i=1}^{n_2} y_i - \frac{n_2}{n_1+n_2} \left( \sum_{i=1}^{n_1} y_i + \sum_{i=1}^{n_2} y_2 \right) \right]}{\frac{n_1+n_2+n_2-n_2}{(n_1+n_2)^2}}$$

see next pg  
for  $\hat{\beta}_0$

$$\hat{\beta}_1 = \frac{\cancel{(n_1+n_2)}}{(n_1+n_2)} \cdot \left[ 1 - \frac{n_2}{n_1+n_2} \right] \sum_{i=1}^{n_1} y_i - \frac{n_2}{n_1+n_2} \sum_{i=1}^{n_1} y_i$$

$$\hat{\beta}_1 = \frac{\frac{n_1+n_2-n_2}{n_1+n_2} \sum_{i=1}^{n_2} y_i - \frac{n_2}{n_1+n_2} \sum_{i=1}^{n_1} y_i}{\frac{n_1 n_2}{(n_1+n_2)}}$$

$\hat{\beta}_1 = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i - \frac{1}{n_1+n_2} \sum_{i=1}^{n_1} y_i$

where  $2 =$

$$\hat{\beta}_1 = \frac{n_1}{n_1+n_2} \times \frac{n_1+n_2}{n_1 n_2} \times \sum_{i=1}^{n_2} y_i - \frac{n_1+n_2}{n_1+n_2} \times \frac{n_2}{n_1+n_2} \times \sum_{i=1}^{n_1} y_i$$

To find  $\hat{\beta}_0$  we do a similar procedure

$$\rightarrow \hat{\beta}_0 \text{ OLS} = \bar{y} - b_1 \bar{x}$$

$$\hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^n y_i - (\bar{y}_2 - \bar{y}_1) \frac{n_2}{n_1 + n_2}$$

$n = n_1 + n_2$

$$\hat{\beta}_0 = \left( \frac{1}{n_1 + n_2} \right) \left[ \sum_{i=1}^{n_1} y_i + \sum_{i=1}^{n_2} y_i \right] - \left[ \frac{n_2}{n_1 + n_2} \right] (\bar{y}_2 - \bar{y}_1)$$

$$\hat{\beta}_0 = \left( \frac{1}{n_1 + n_2} \right) \left[ \sum_{i=1}^{n_1} y_i + \sum_{i=1}^{n_2} y_i - n_2 \left( \frac{1}{n_2} \sum_{i=1}^{n_2} y_i - \frac{1}{n_1} \sum_{i=1}^{n_1} y_i \right) \right]$$

$$\hat{\beta}_0 = \left( \frac{1}{n_1 + n_2} \right) \left[ \sum_{i=1}^{n_1} y_i + \frac{n_2}{n_1} \sum_{i=1}^{n_1} y_i + \sum_{i=1}^{n_2} y_i - \sum_{i=1}^{n_2} y_i \right]$$

$$\hat{\beta}_0 = \left( \frac{1}{n_1 + n_2} \right) \left[ (1 + \frac{n_2}{n_1}) \sum_{i=1}^{n_1} y_i \right]$$

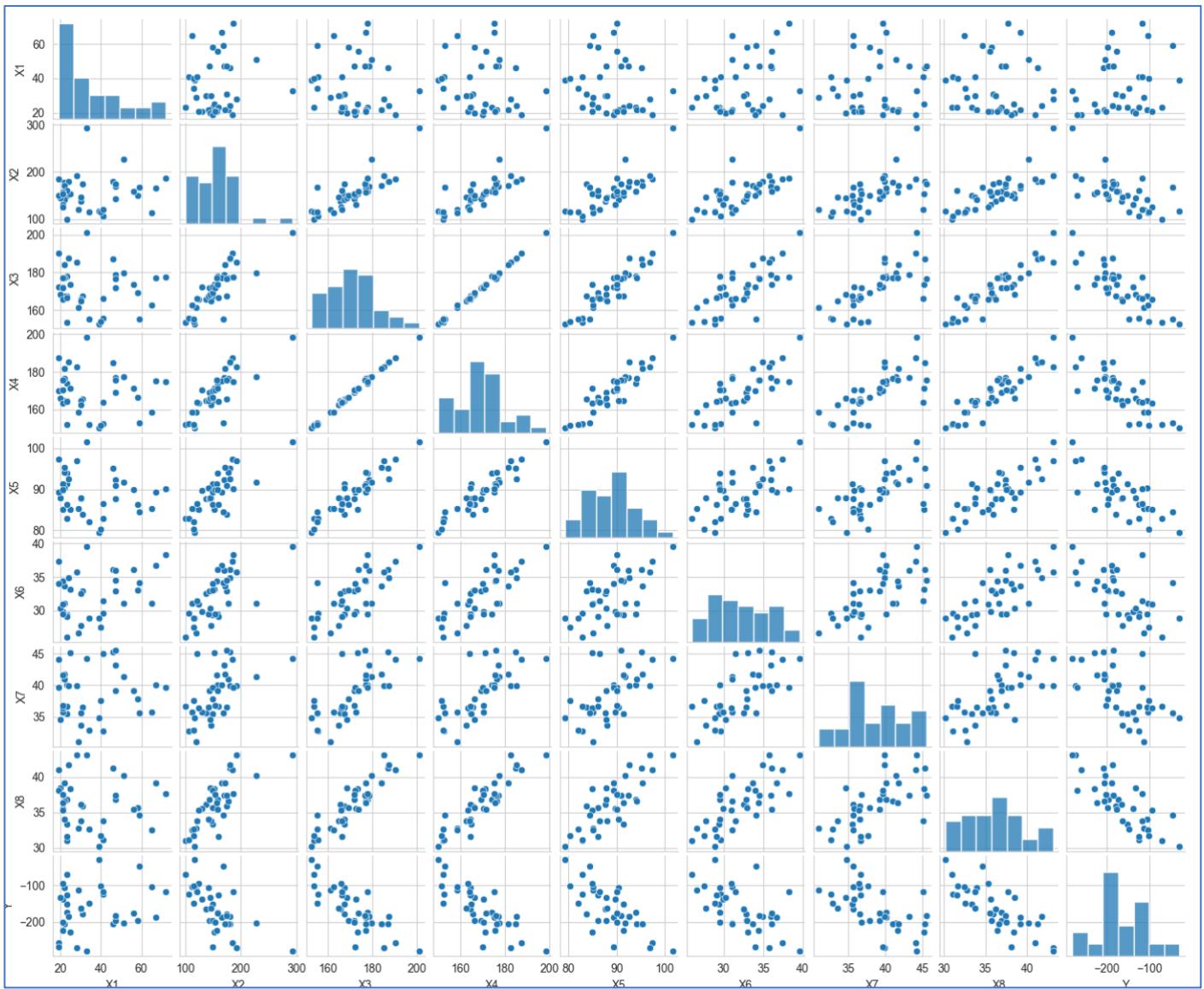
$$\hat{\beta}_0 = \frac{n_1 + n_2}{n_1} \times \frac{1}{n_1 + n_2} \times \sum_{i=1}^{n_1} y_i$$

$$\boxed{\hat{\beta}_0} = \cancel{\frac{n_1 + n_2}{n_1 + n_2}} \times \frac{1}{n_1} \sum_{i=1}^{n_1} y_i = \boxed{\bar{y}_1} \quad \checkmark$$

$$\therefore \begin{bmatrix} \hat{\beta}_0 = \bar{y}_P \\ \hat{\beta}_1 = \bar{y}_T - \bar{y}_P \end{bmatrix}$$

## Question 2)

Q2 a) Pairs plot to assess Multicollinearity



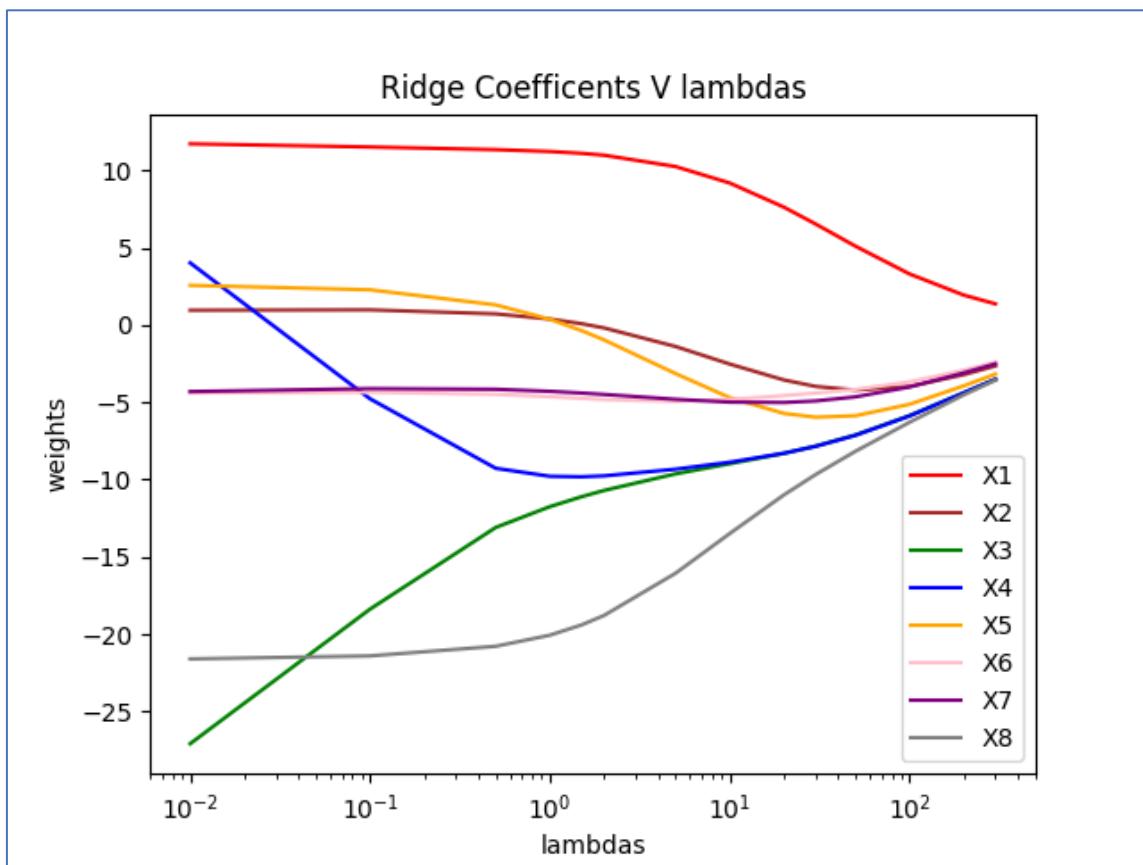
From the plot above we can see that many of the features are highly correlated e.g. (x4,x3), (x8,x3) and (x4,x5). Typically, a feature's coefficient represents the effect of a 1-unit change in that variable on the response while holding all other variables constant. However as seen above a very strong correlation between variables (e.g. x3,x4) it becomes difficult for the linear model to estimate the relationship between each feature and the response *independently* because the features tend to change in unison. So we have an imprecise estimate of the effect of independent changes in the features. **A model with multicollinear data has very high variance but very low bias which results in overfitting.**

## Q2 b)

As we can see below the sum of squares is  $\approx 38$ . With mean  $\approx 0$  and var 1 for each factor.

| ----mean---- |                | ----variance---- |                | ----sum of sq's---- |                   |
|--------------|----------------|------------------|----------------|---------------------|-------------------|
| X1           | -2.921640e-17  | X1               | 1.000000       | X1                  | 38.0              |
| X2           | 2.859555e-16   | X2               | 1.000000       | X2                  | 38.00000000000001 |
| X3           | -1.602519e-15  | X3               | 1.000000       | X3                  | 38.00000000000002 |
| X4           | 1.804112e-16   | X4               | 1.000000       | X4                  | 37.99999999999998 |
| X5           | 1.845015e-15   | X5               | 1.000000       | X5                  | 38.0              |
| X6           | -1.051790e-16  | X6               | 1.000000       | X6                  | 38.0              |
| X7           | 9.714451e-17   | X7               | 1.000000       | X7                  | 38.00000000000014 |
| X8           | -1.234393e-16  | X8               | 1.000000       | X8                  | 37.99999999999986 |
| Y            | -1.648849e+02  | Y                | 58.857318      |                     |                   |
|              | dtype: float64 |                  | dtype: float64 |                     |                   |

## Q2 c) Graph produced mapping the change in coefficient values as Lambda values change

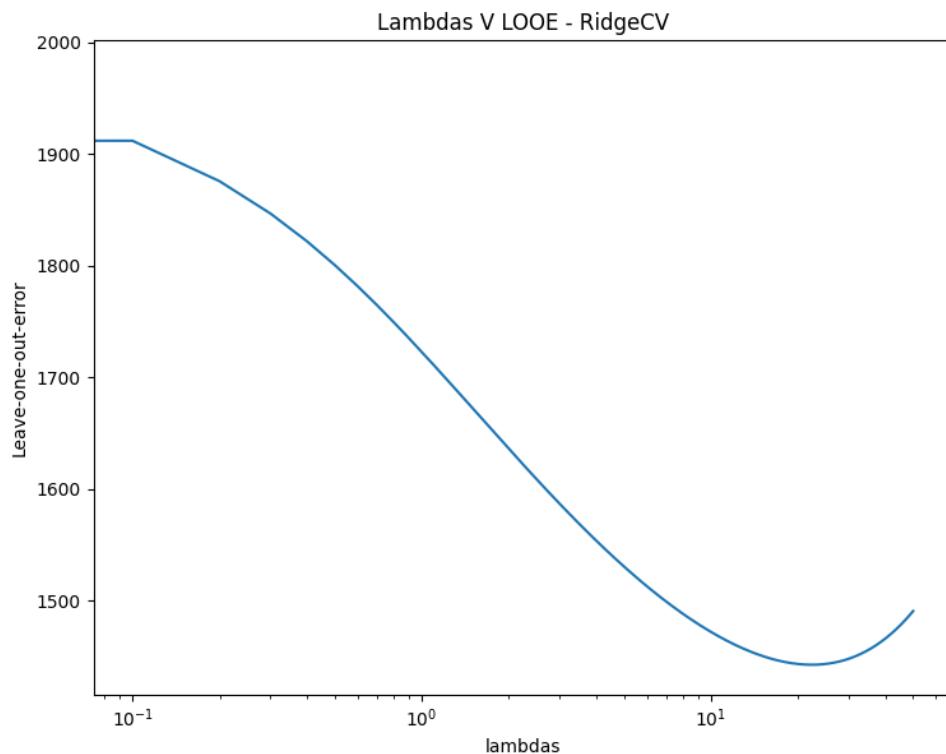


From the plot as alpha increases the coefficients start to converge to 0. This is the power of ridge regression, making the coefficients smaller to limit the collinearity between predictors. For regressor 3 it completely flipped the sign on the coefficient, and it converged to a smaller value. Although regressor 4 remained the same sign the change in magnitude for higher lambda values was significant. Regressor 5 remained similar. Also, the coefficient of regressor X1 continues to remain important even with increasing penalty.

Code used to produce plot in Q2 c)

```
105 # part c - Ridge regression Plots
106 # summarize shape
107 print(normalized_df.shape)
108 y = normalized_df['Y']
109 X = normalized_df.drop('Y',axis=1)
110
111 # get our coefficents for each lambda
112 lambdas = np.array([0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300])
113 ridge = Ridge()
114 coefs = []
115 for a in lambdas:
116     ridge.set_params(alpha = a)
117     ridge.fit(X, y)
118     coefs.append(ridge.coef_)
119
120 print(coefs)
121
122 # plot
123 ax = plt.gca()
124 labels = ['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8']
125 color_cycle= ['red', 'brown', 'green', 'blue', 'orange', 'pink', 'purple', 'grey']
126 for i in range(len(coefs[0])):
127     ax.plot(lambdas,[pt[i] for pt in coefs], color=color_cycle[i],label = f"{labels[i]}")
128
129 ax.set_xscale('log')
130 ax.autoscale()
131 plt.xlabel('lambdas')
132 plt.ylabel('weights')
133 plt.title('Ridge Coefficents V lambdas')
134 plt.legend()
135 plt.show()
```

Q2 d) Graph produced through LOOCV -Ridge



The LOOE from OLS = 1975.414739342173  
 Minimum LOOE from ridge = 1442.6982227952915  
 Best lambda = 22.3

We can see from the graph that the Error gradually reduces to a minimum at lambda = 22.3 and then slowly rises again. Compared to OLS which has a MSE of 1975.4 the ridge regression produces an error of 1442.69 at its best. This matches what we know about ridge regression as it produces estimators that have inherently less variance than OLS but are a slightly biased

Code used to produce the above results in Q2d) :

```

230 #part d LOOCV from scratch
231 #find optimal lambda
232 lambdas = np.arange(start=0, stop=50.1, step=0.1)
233 copy_X = normalized_df.drop('Y',axis=1)
234 copy_y = normalized_df['Y']
235
236 ridge_d = Ridge()
237 row_it = 0
238 mse_for_lambdas = []
239
240 for a in lambdas:
241     sq_errors = []
242     while row_it < 38: # Note : one observation is equal to one row per defintion of MLR
243         dropped_row_X = copy_X.loc[row_it]
244         dropped_y = copy_y.iloc[row_it]
245
246         #remove the ith obs
247         ith_removed_X = copy_X.drop(row_it, axis = 0)
248         ith_removed_y = copy_y.drop(row_it, axis = 0)
249         ridge_d.set_params(alpha = a) # set to current lambda
250         ridge_d.fit(ith_removed_X, ith_removed_y)
251         dropped_row_X = dropped_row_X.values.reshape(1, -1)
252
253         #get prediction error
254         prediction = ridge_d.predict(dropped_row_X)
255         prediction_error = pow((dropped_y - prediction[0]),2)
256         sq_errors.append(prediction_error)
257         row_it= row_it + 1
258
259     row_it = 0
260     mse = (sum(sq_errors))/38 # find the Leave one out error average for lambda = a
261     mse_for_lambdas.append(mse)
262
263 #find best lambda
264 minMSE = min(mse_for_lambdas)
265 minMSE_index = mse_for_lambdas.index(min(mse_for_lambdas))
266 best_lambda = lambdas[minMSE_index]
267 print(f"MIN mse = {minMSE} -----")
268 print(f"best lambda = {best_lambda} -----")
269
270 #plot
271 ax = plt.gca()
272 ax.set_xscale('log')
273 ax.autoscale()
274 plt.xlabel('lambdas')
275 plt.ylabel('Leave-one-out-error')
276 plt.title('Lambdas V LOOE - RidgeCV')
277 plt.plot(lambdas, mse_for_lambdas, color ="tab:blue")
278 plt.show()
```

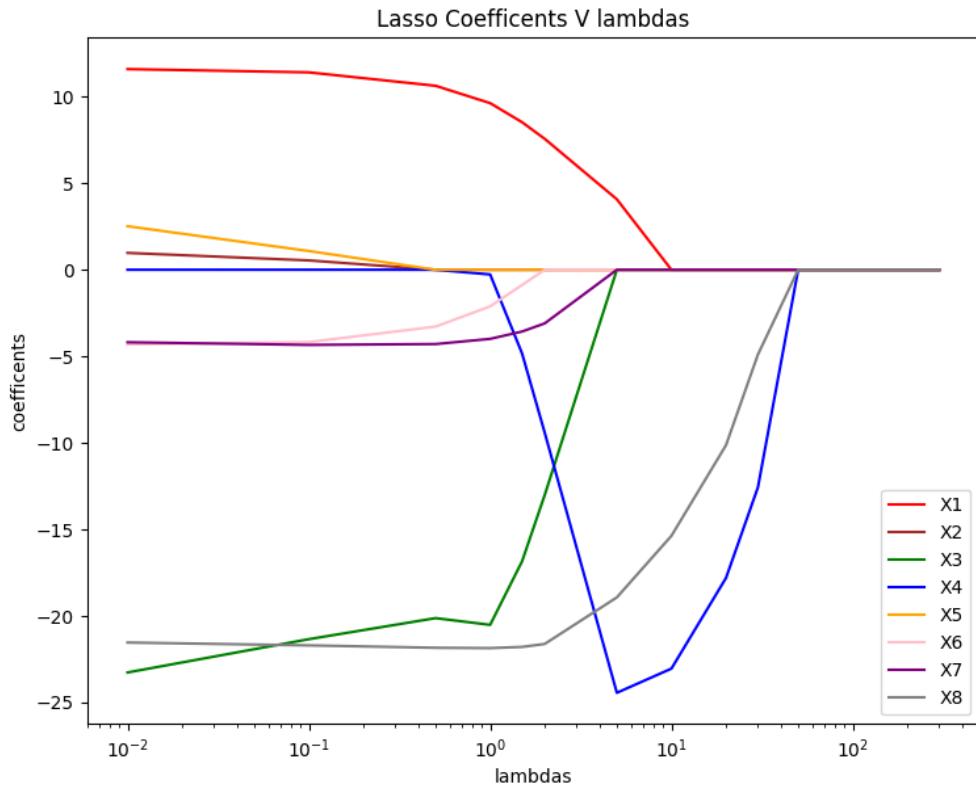
```

280 ols = LinearRegression()
281 sq_errors = []
282 # LOOCV for OLS
283 while row_it < 38:
284     dropped_row_X = copy_X.loc[row_it]
285     dropped_y = copy_y.iloc[row_it]
286     ith_removed_X = copy_X.drop(row_it, axis = 0)
287     ith_removed_y = copy_y.drop(row_it, axis = 0)
288
289     ols.fit(ith_removed_X, ith_removed_y)
290     dropped_row_X = dropped_row_X.values.reshape(1, -1)
291
292     prediction = ols.predict(dropped_row_X)
293     prediction_error = pow((dropped_y - prediction[0]), 2)
294     sq_errors.append(prediction_error)
295     row_it = row_it + 1
296
297 mse = (sum(sq_errors))/38
298 print(f"the LOOE from OLS is {mse}")

```

Code used to produce results for OLS comparison in Q2d)

Q2e) Graph produced mapping the change in coefficient values as Lambda values change

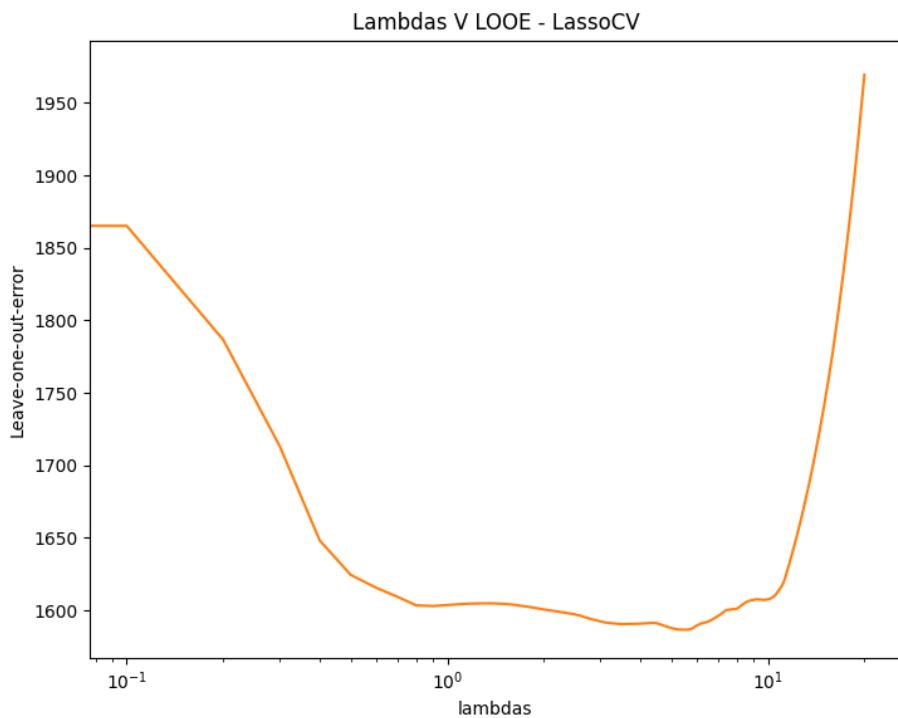


As seen the above graph compared to the Ridge Regression done in part c) there a stark difference present between the path coefficient's take with respect to high penalties. From the graph all coefficients go to 0 at high enough lambdas. The coefficient on X1 takes an expect path and remains important While the coefficient on X4 is initially 0 then a has a significant negative affect while rising to zero again. Indicating further investigation between X4 and y is needed.

Code used to produce above results for Q2e)

```
209 #Q2 part e
210 #get our lambdas
211 lambdas = np.array([0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 30, 50, 100, 200, 300])
212
213 copy_X = normalized_df.drop('Y',axis=1)
214 copy_y = normalized_df['Y']
215 lasso = Lasso()
216 coefs = []
217
218 for a in lambdas:
219     lasso.set_params(alpha = a)
220     lasso.fit(copy_X, copy_y)
221     coefs.append(lasso.coef_)
222 #sanity check
223 print(coefs)
224 #plot
225 ax = plt.gca()
226 labels = ['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8']
227 # ax.plot(lambdas, coefs)
228 color_cycle= ['red', 'brown', 'green', 'blue', 'orange', 'pink', 'purple', 'grey']
229 for i in range(len(coefs[0])):
230     ax.plot(lambdas,[pt[i] for pt in coefs], color=color_cycle[i],label = f'{labels[i]}')
231
232 ax.set_xscale('log')
233 ax.autoscale()
234 plt.axis('tight')
235 plt.xlabel('lambdas')
236 plt.ylabel('coefficients')
237 plt.title('Lasso Coefficients V lambdas')
238 plt.legend()
239 plt.show()
```

Q2 f) Graph produced from LOOCV – Lasso



```
MIN LOOE = 1586.6715081806428
best lambda = 5.5
```

From the graph we can see that like Ridge regression the error drops significantly for 'medium' levels of lambda but dramatically increases after a certain point.

Code used to the above results for Q2 f)

```
245 #Q2 part f
246 lambdas = np.arange(start=0, stop=20.1, step=0.1)
247 copy_X = normalized_df.drop('Y', axis=1)
248 copy_y = normalized_df['Y']
249
250 lasso = Lasso()
251 row_it = 0
252 mse_for_lambdas = []
253
254 for a in lambdas:
255     sq_errors = []
256
257     while row_it < 38:
258         dropped_row_X = copy_X.loc[row_it]
259         dropped_y = copy_y.iloc[row_it]
260         #remove the ith obs
261         ith_removed_X = copy_X.drop(row_it, axis = 0)
262         ith_removed_y = copy_y.drop(row_it, axis = 0)
263
264         #Predict
265         lasso.set_params(alpha = a) # set to current lambda
266         lasso.fit(ith_removed_X, ith_removed_y)
267         dropped_row_X = dropped_row_X.values.reshape(1, -1)
268         prediction = lasso.predict(dropped_row_X)
269         prediction_error = pow((dropped_y - prediction[0]),2)
270         sq_errors.append(prediction_error)
271
272         row_it+= 1
273
274     row_it = 0
275     mse = (sum(sq_errors))/38
276     mse_for_lambdas.append(mse)
277 # Find minimum LOOE for a lambda
278 minMSE = min(mse_for_lambdas)
279 minMSE_index = mse_for_lambdas.index(min(mse_for_lambdas))
280 best_lambda = lambdas[minMSE_index]
281 print(f"MIN mse = {minMSE} -----")
282 print(f"best lambda = {best_lambda} -----")
283 #plot graph
284 ax = plt.gca()
285 ax.set_xscale('log')
286 ax.autoscale()
287 plt.xlabel('lambdas')
288 plt.ylabel('Leave-one-out-error')
289 plt.title('Lambdas V LOOE - LassoCV')
290 plt.plot(lambdas, mse_for_lambdas, color ="tab:orange")
291 plt.show()
```

## Q2 g)

The differences between ridge and lasso regression lie in the minimisation criterion and specifically we can observe how the different criteria have affected our predictions. For our case the minimum error for ridge regression produced was 1442 compared to 1586 for lasso found through out of sample prediction. **So, Ridge regression would be the preferred model for prediction.**

Also, our data was shown to be highly correlated and gave little reason to choose between different linear combinations of colinear predictors. Ridge regression will perform better as it tends to shrink the “group” (the correlated variables) proportionately reducing variance. Whereas Lasso doesn’t since it promotes setting individual regression coefficients to zero to reduce model size.

Q 3)

show

$$a) |\langle y, x_B \rangle| \leq \max_j |x^T y| \sum_i |B_j|$$

$$\begin{aligned} \text{LHS: } |\langle y, x_B \rangle| &= |\langle x_B, y \rangle| \\ &= |\langle B, x^T y \rangle| \\ &= \langle |\langle x^T y \rangle|, |B| \rangle \end{aligned}$$

→ let  $x^T y = \alpha$  where  $\alpha$  is a vector.

$$\text{hence LHS} = \langle |\alpha|, |B| \rangle$$

note  
 $B \in \mathbb{R}^p$   
 $x^T y \in \mathbb{R}^p$   
as  $x^T \in \mathbb{R}^{p \times n}$   
 $y \in \mathbb{R}^n$

$$\begin{aligned} &= \left\langle x^T \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{pmatrix}, \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{pmatrix} \right\rangle \\ &= |\beta_1 \alpha_1 + \beta_2 \alpha_2 + \dots + \beta_p \alpha_p| \\ &\leq |\beta_1 \alpha_j + \beta_2 \alpha_j + \dots + \beta_p \alpha_j| \\ &\quad \text{where } \alpha_j = \max_j (x^T y) \\ &= |\alpha_j (\beta_1 + \beta_2 + \dots + \beta_p)| \\ &= |\alpha_j \sum_i \beta_i| \\ &= \max_j |x^T y| \sum_i |B_j| \\ &= \text{RHS.} \end{aligned}$$

→ Hence LHS  $\leq$  RHS

$$b) \text{ assume } \lambda \geq \max_j |x^T y|$$