Q1.

```
Coefficients:
            Estimate   Pr(>|t|)← p value
(Intercept)  6.08283    <2e-16
data_1$x     2.23236    <2e-16

Multiple R-squared:  0.9631,   Adjusted R-squared:  0.9627
```

Correlation coefficient between y-predicted and y given is 0.9813566

The R Code for Q1 is below:

```
data_1 = read.table("E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/130010009_Viv
ek_Assignment1/a1-data-set.csv",header = TRUE, sep = ",")

lr_initial = lm(data_1$y ~ data_1$x)
list_null = (is.na(data_1$y))
data_1$y[list_null] = summary(lr_initial)$coefficients[2,1]*data_1$x[list_null]
+ summary(lr_initial)$coefficients[1,1]

lr_final = lm(data_1$y ~ data_1$x)
summary(lr_final)
ypred = predict(lr_final)
cor_1 = cor(ypred,data_1$y)
```

Q2.

```
Coefficients:
            Estimate Pr(>|t|)← pvalue
(Intercept)  6.08283    <2e-16
x_1          2.23236    <2e-16

Multiple R-squared:  0.9537,   Adjusted R-squared:  0.9531
```

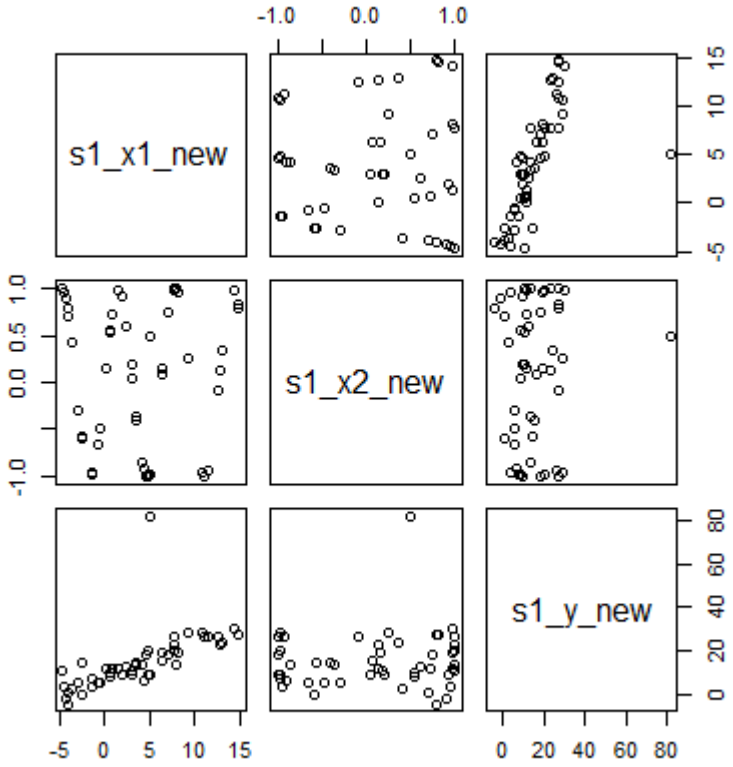
Correlation coefficient between y-predicted and y given is 0.9765599

The R code for Q2 is below:

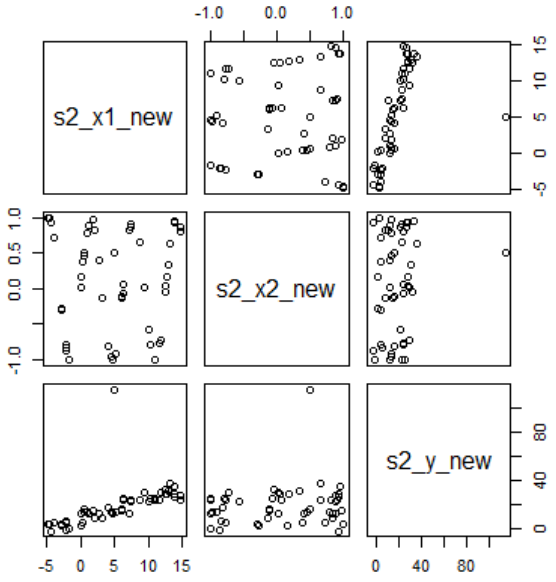
```
data_2 = read.table("E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/130010009_Viv
ek_Assignment1/a1-data-set.csv",header = TRUE, sep = ",")

list_null = (is.na(data_1$y))
y_1 = data_1$y[!list_null]
x_1 = data_1$x[!list_null]
lr_initial = lm(y_1 ~ x_1)
summary(lr_initial)
ypred = predict(lr_initial)
cor2 = cor(ypred,y_1)
```

Q3. The above results indicate that the least square line is same for both the cases i.e when values are imputed using least square regression on the remainin g dataset, and then fitting a new line on the imputed dataset & when the missin g data is removed.

Q4.

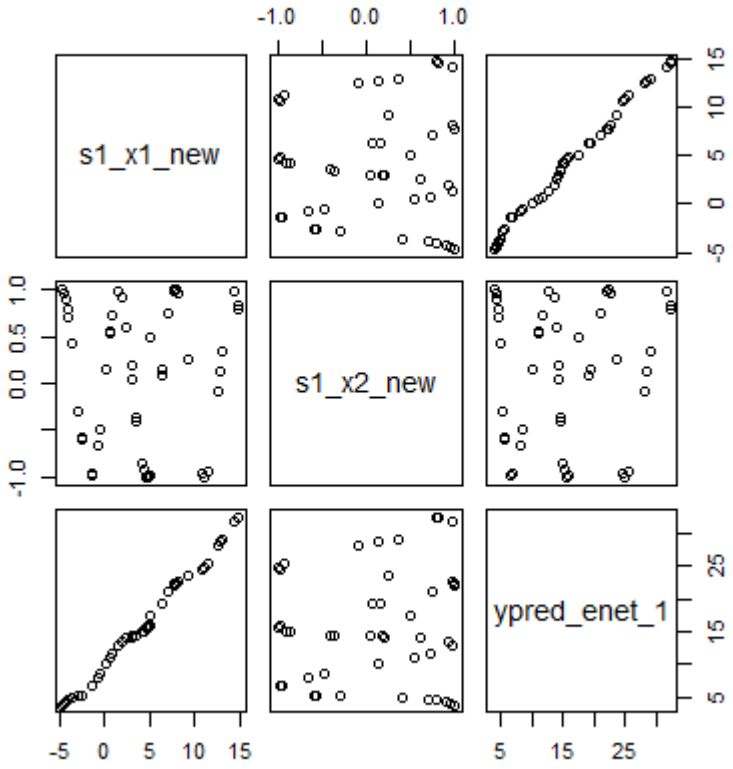


Plot of s1 data

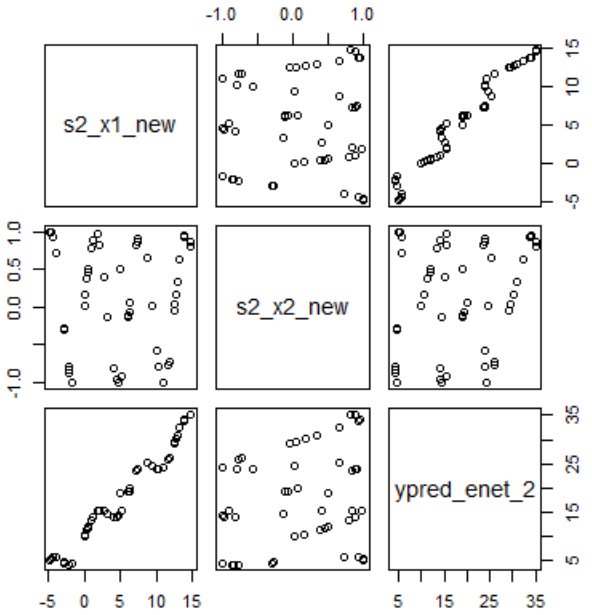


Plot of s2 data

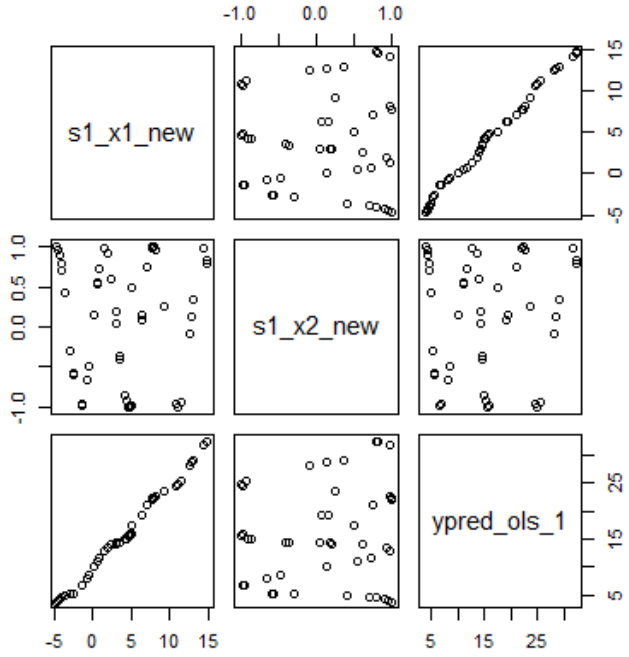
After running ols and enet on the new s1 and s2 dataset(includes the outlier point), these are the plots of the y_predicted values of ols and enet with x1 and x2.



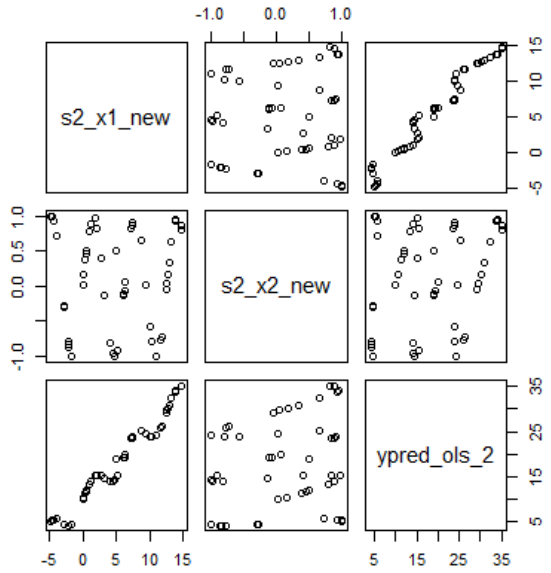
Enet on s1 dataset



ENet on s2 dataset



OLS on s1 dataset



OLS on s2 dataset

As can be clearly seen, OLS deviates sharply due to the outlier point, compared to enet. It can be seen from the top right graphs of each plot, where in OLS, there are many curvy paths and shifts drastically due to the outlier. Elastic net has lesser variance compared to OLS.

The R code for Q3 is given below:

```
quiz2_q3=read.csv("E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/a2-data-set.csv",header=TRUE)

x1 = quiz2_q3[1:1000,1]

x2 = quiz2_q3[1:1000,2]

y = quiz2_q3[1:1000,3]

mydf = data.frame(x1,x2,y)


set.seed(8)

sindex =sample.int(1000,50)

s1index = sample.int(1000,50)
```

```
mydf.s1 = data.frame(x1 = mydf$x1[sindex],x2 = mydf$x2[sindex], y=mydf$y[sindex])

mydf.s2 = data.frame(x1 = mydf$x1[s1index],x2 = mydf$x2[s1index], y=mydf$y[s1index])


y_outlier1 = 7*IQR(mydf.s1$y)

y_outlier2 = 7*IQR(mydf.s2$y)

x1_outlier = 5.0

x2_outlier = 0.5

s1_x1_new = c(mydf.s1$x1,x1_outlier)

s1_x2_new = c(mydf.s1$x2,x2_outlier)

s2_x1_new = c(mydf.s2$x1,x1_outlier)

s2_x2_new = c(mydf.s2$x2,x2_outlier)

s1_y_new = c(mydf.s1$y,y_outlier1)

s2_y_new = c(mydf.s2$y,y_outlier2)

plot(data.frame(s1_x1_new,s1_x2_new,s1_y_new))

plot(data.frame(s2_x1_new,s2_x2_new,s2_y_new))

tdf1 = data.frame(s1_x1_new,s1_x2_new)

tdf2 = data.frame(s2_x1_new,s2_x2_new)

ctrl = trainControl(method="CV", number=10, returnResamp = "all")


tr.lm1 = train(as.matrix(tdf1),s1_y_new , method="lm", trControl=ctrl)

tr.enet1 = train(as.matrix(tdf1),s1_y_new , method="enet", trControl=ctrl)


tr.lm2 = train(as.matrix(tdf2),s2_y_new , method="lm", trControl=ctrl)

tr.enet2 = train(as.matrix(tdf2),s2_y_new , method="enet", trControl=ctrl)

ypred_enet_1 = predict(tr.enet1)
```

```
ypred_enet_2 = predict(tr.enet2)

ypred_ols_1 = predict(tr.lm1)

ypred_ols_2 = predict(tr.lm2)

plot(data.frame(s1_x1_new,s1_x2_new,ypred_enet_1))

plot(data.frame(s1_x1_new,s1_x2_new,ypred_ols_1))

plot(data.frame(s2_x1_new,s2_x2_new,ypred_enet_2))

plot(data.frame(s2_x1_new,s2_x2_new,ypred_ols_2))

mydf1 = data.frame(s1_x1_new,s1_x2_new,s1_y_new)

mydf2 = data.frame(s2_x1_new,s2_x2_new,s2_y_new)

write.csv(mydf1,"E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/s1.csv")

write.csv(mydf2,"E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/s2.csv")
```

Q4.

For dataset s3:

```
Multiple R-squared:  0.9331,   Adjusted R-squared:  0.9302
```

For dataset s4:

```
Multiple R-squared:  0.643,    Adjusted R-squared:  0.6275
```

S3 was created using

y3 = y3 + rnorm(25, 0, 3)

and S4 was created using

y4 = y4 + rnorm(25,3,5)

for the base function y = 3*x + 5

The logic behind it was that with increase in variance, there will be more outliers , and since OLS is sensitive to outliers, the least fit line deviates away from the base function line to accommodate or go as near as possible to the outlier. Hence, the R squared , which is the ratio of the error due to regression upon total error, increases.

The R code for Q4 is below:

```
set.seed(1)
```

```
x = runif(25, 1, 10)

y3 = 3 * x + 5

y4 = 3 * x + 5


y3 = y3 + rnorm(25, 0, 3)

y4 = y4 + rnorm(25,3,5)

mydf3 = data.frame(x, y3)

mydf4 = data.frame(x,y4)


lm3 = lm(y3~x)

lm4 = lm(y4~x)

write.csv(mydf3,"E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/s3.csv")

write.csv(mydf4,"E:/Acad_9th_Sem/ME781_Data_Mining/Assignment/s4.csv")
```