Name: Sumit Singh

Roll No: 380

Class: TYBSC CS A

Subject: Data Science

PRACTICAL 2

Aim: Simple /Multiple Linear Regressions.

#IMPOER DATASET:

Command:

>data=read.csv ("D://tycs/score.csv")

>data

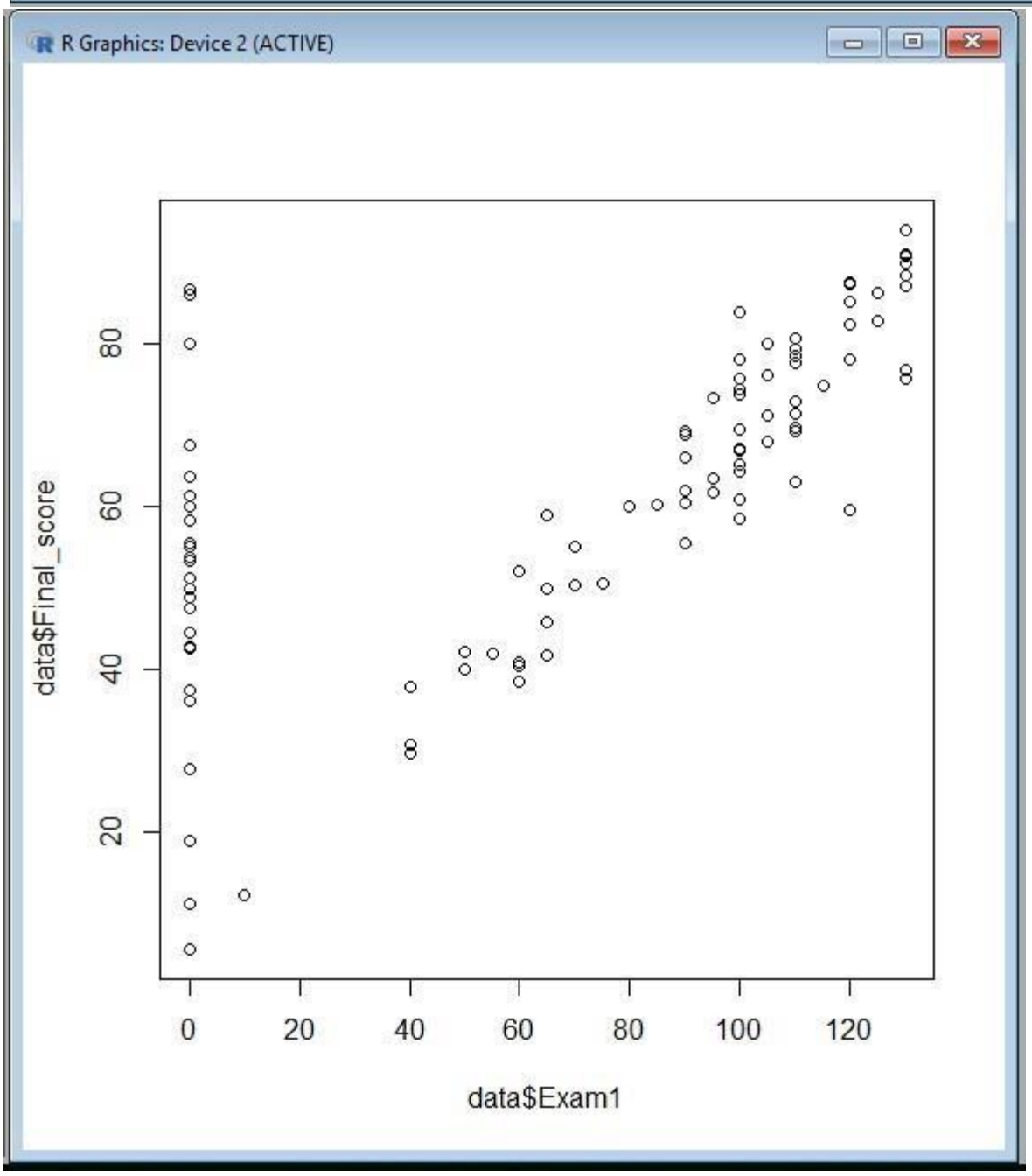
```
> data= read.csv("D://TYCS21/score.csv")
> data
   Exam1 Exam2 Exam3 Exam4 Final_score Grade
1     60    10    16   7.0       40.79     C
2     90     0     0   0.0       69.23     B
3    130    20    24   1.0       76.75     B
4    130    10    24   8.5       75.66     B
5     90     5    22   9.5       55.48     C
6    100    30    20   3.0       67.11     B
7    105    20    22   8.0       67.98     B
8    120    40    18  16.0       85.09     A
9    120    20    30  18.0       82.46     A
10   130    45    22  10.5       91.01     A
11    90    40    20   7.0       68.86     B
12   130    30    28  10.5       87.06     A
13   100    30    22   6.5       69.52     B
14     0    30    18   0.0       60.00     B
15     0    30    18   0.0       60.00     B
16    80     0    24   3.0       60.11     B
17   105    40    22   6.5       76.10     B
18    10     0     0   8.0       12.16     D
19   130    35    24   0.0       90.00     A
20     0    15    20   7.0       42.86     C
21    40    10    14   6.0       30.70     D
22    90    15    28   8.5       62.06     B
23   110     0    24   9.5       80.62     A
24    65     5    24   1.0       41.67     C
25    55    15    18   0.0       41.90     C
26   100    50    30  11.5       83.99     A
27    95    40    24   8.0       73.25     B
28     0    10    24   0.0       42.50     C
29     0     0    18   0.0       60.00     B
30    65    20    20   0.0       50.00     C
31   110    25    18   6.0       69.74     B
32   130    45    24   8.0       90.79     A
33   120    40    30   9.0       87.28     A
```

#PLOT THE DATASET:

COMMAND:

>plot(x=data$Exam1,y=data$Final_score)

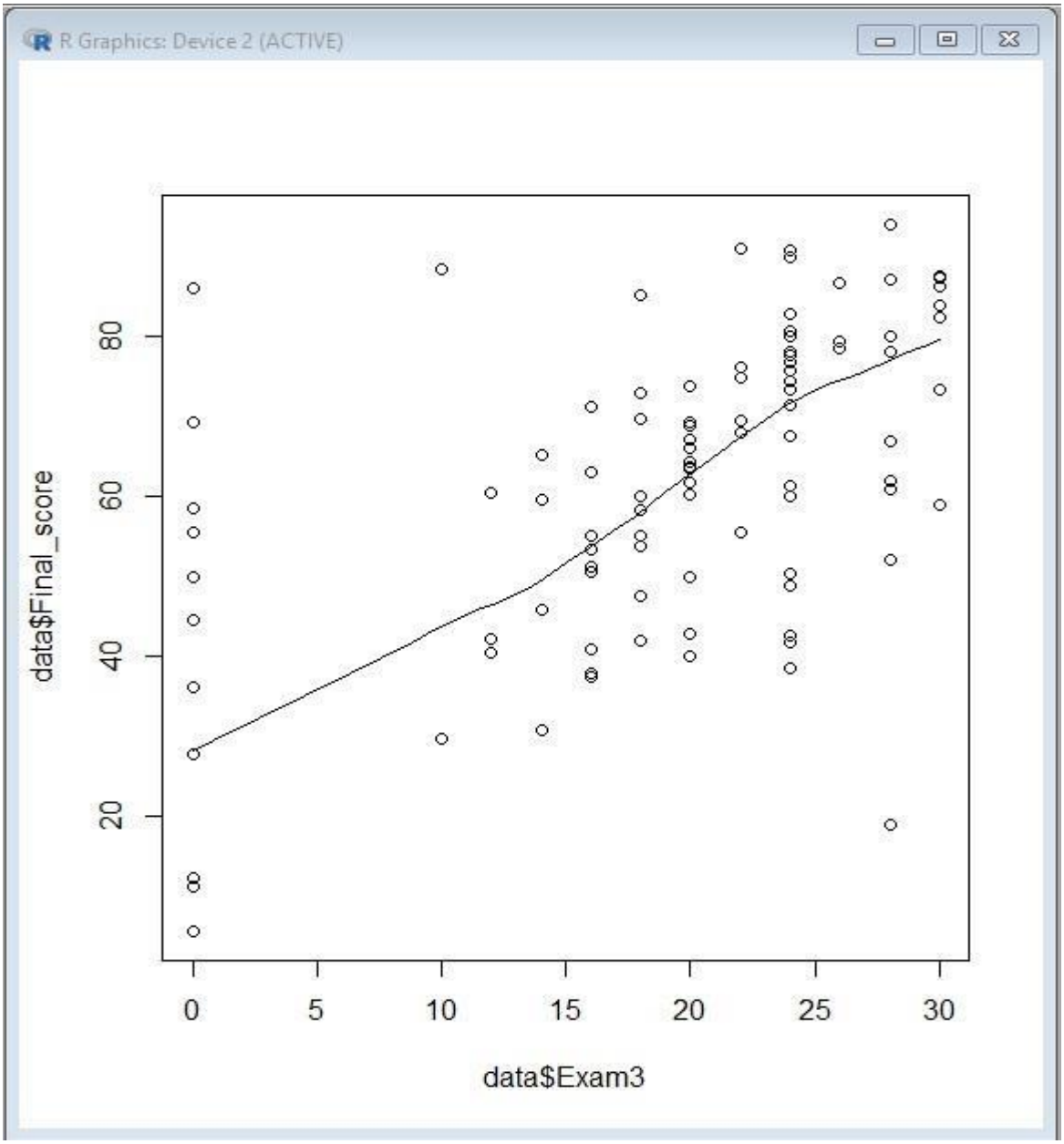
```
> plot(x=data$Exam1,y=data$Final_score)
>
```



#PLOT THE SCATTER DIAGRAM:

```
>scatter.smooth(x=data$Exam3,y=data$Final_score)
```

```
> scatter.smooth(x=data$Exam3,y=data$Final_score)
>
```




```
> cor(data$Exam3,data$Final_score)
[1] 0.6046352
>
```

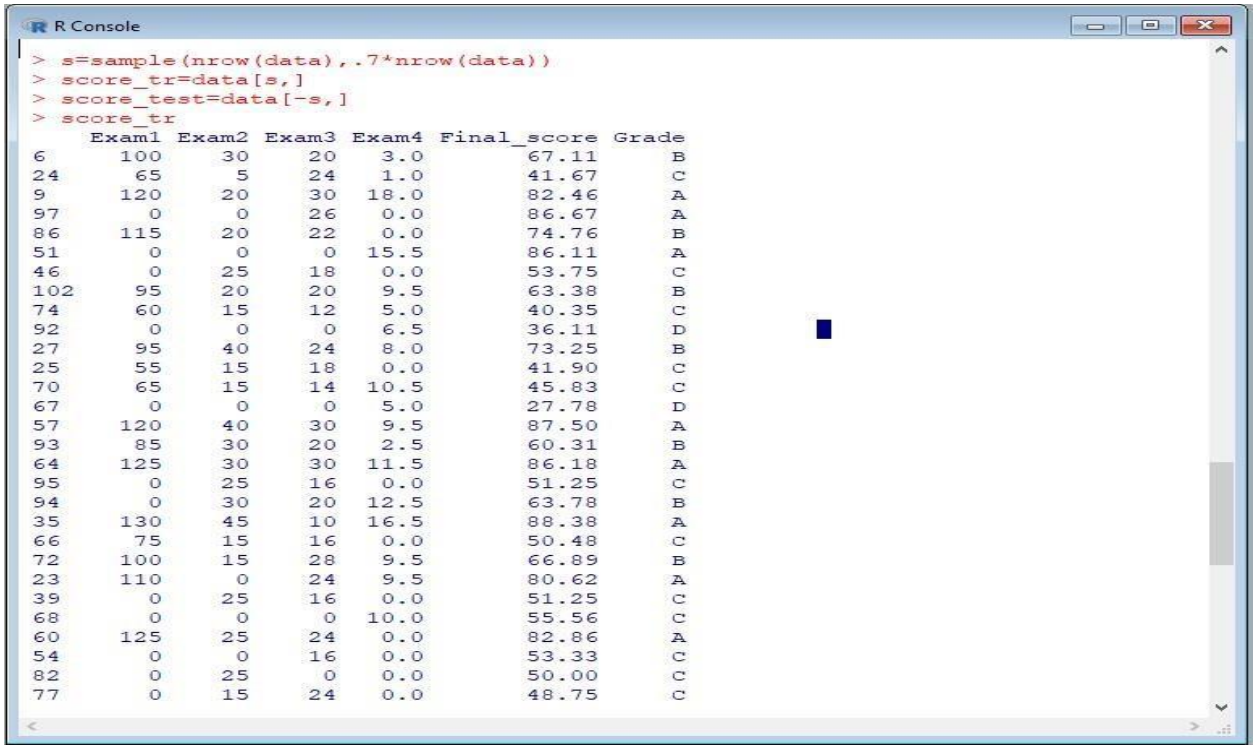
#PARTITIONING THE DATABASE INTO TRAINING AND TESTING SET

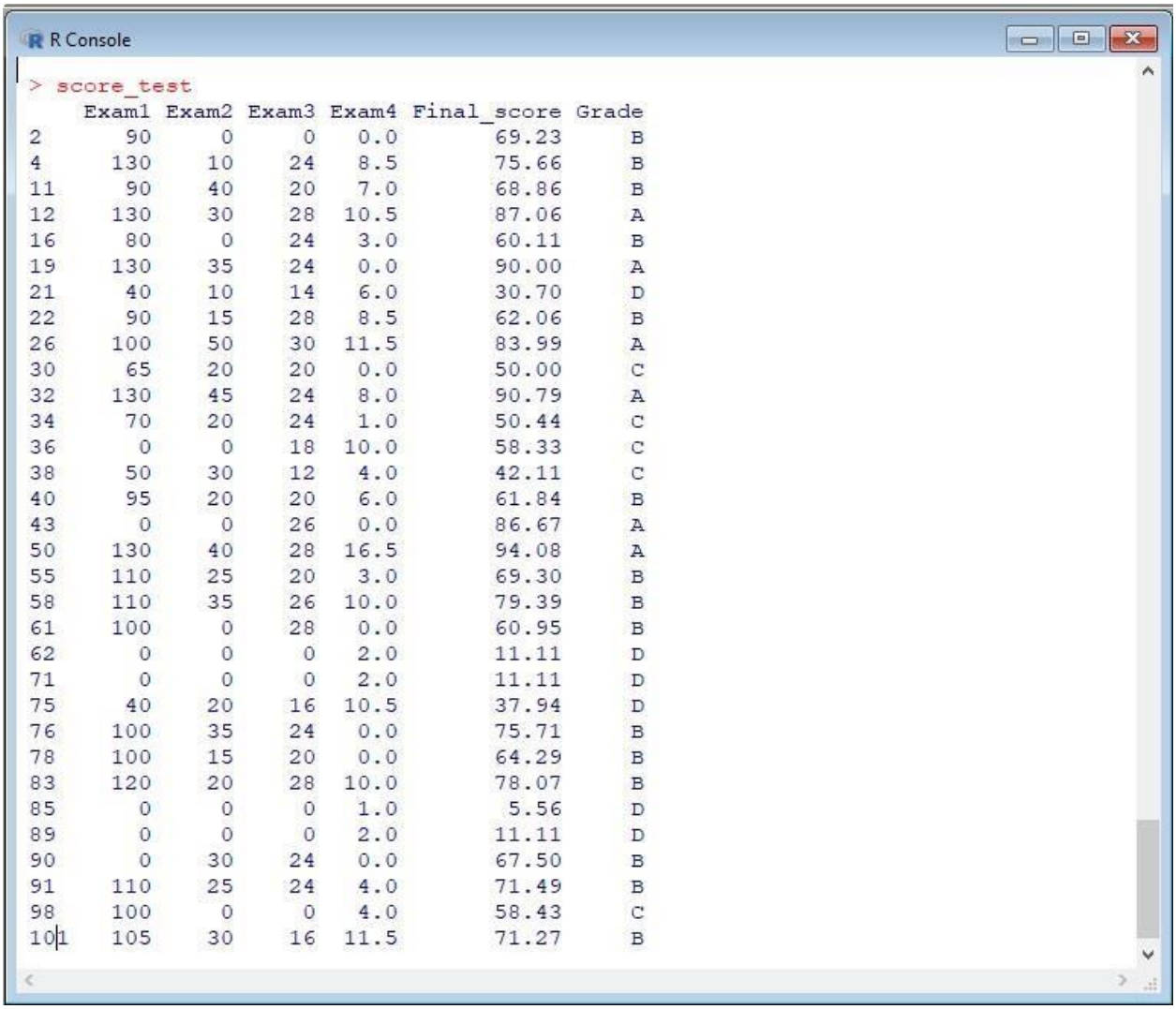
>s=sample(nrow(data),.7*nrow(data))

>score_tr=data[s,]

>score_test=[-s,]

Score_tr






```
> linmon=lm(Final_score~Exam3,data=score_tr)
> print(linmod)
Error in print(linmod) : object 'linmod' not found
> print(linmon)

Call:
lm(formula = Final_score ~ Exam3, data = score_tr)

Coefficients:
(Intercept)        Exam3
     39.537        1.119

> 
```

```
> pdata=predict(linmon,score_test)
> summary(linmon)

Call:
lm(formula = Final_score ~ Exam3, data = score_tr)

Residuals:
    Min      1Q  Median      3Q     Max
-52.005  -9.967   1.666  10.500  46.573

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.5367     4.8090   8.221 7.15e-12 ***
Exam3         1.1189     0.2362   4.737 1.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.42 on 70 degrees of freedom
Multiple R-squared:  0.2427,    Adjusted R-squared:  0.2319
F-statistic: 22.44 on 1 and 70 DF,  p-value: 1.101e-05

>
```

#CREATING A MODEL

```
> actual_predict=data.frame(cbind(actuals=score_test$Final_score,predicteds=pdata))
> actual_predict
   actuals predicteds
2    69.23   39.53669
4    75.66   66.38965
11   68.86   61.91416
12   87.06   70.86515
16   60.11   66.38965
19   90.00   66.38965
21   30.70   55.20092
22   62.06   70.86515
26   83.99   73.10290
30   50.00   61.91416
32   90.79   66.38965
34   50.44   66.38965
36   58.33   59.67641
38   42.11   52.96317
40   61.84   61.91416
43   86.67   68.62740
50   94.08   70.86515
55   69.30   61.91416
58   79.39   68.62740
61   60.95   70.86515
62   11.11   39.53669
71   11.11   39.53669
75   37.94   57.43867
76   75.71   66.38965
78   64.29   61.91416
83   78.07   70.86515
85    5.56   39.53669
89   11.11   39.53669
90   67.50   66.38965
91   71.49   66.38965
98   58.43   39.53669
```

#PREDICTING THE OUTPUT ON TEST DATASET

```
> cor(actual_predict$actual,actual_predict$predict)
[1] 0.7674963
>
```

```
> mape= mean(abs((actual_predict$predicteds - actual_predict$actual))/ actual_predict$actual)*100
> mape
[1] 60.6191
> mape= mean(abs((actual_predict$predicteds - actual_predict$actual))/ actual_predict$actual)
> mape
[1] 0.606191
>
```

✞ Plot Scatter plot

```
>x=read.csv("D:/TYCS46/score.csv")
>x
   Exam1 Exam2 Exam3 Exam4 Final_score Grade
1     60    10    16   7.0       40.79     C
2     90     0     0   0.0       69.23     B
3    130    20    24   1.0       76.75     B
4    130    10    24   8.5       75.66     B
5     90     5    22   9.5       55.48     C
.
.
.
.
.
.
> s=sample(nrow(x),.7*nrow(x))
>score_tr=x[s,]
>score_test=x[-s,]
>scatter.smooth(x=score_tr$Exam3,y=score_tr$Final_score)
```

✞ Get Linear regression

```
>linmod=lm(Final_score~Exam3,data=score_tr)
>print(linmod)

Call:
lm(formula = Final_score ~ Exam3, data = score_tr)

Coefficients:
(Intercept)       Exam3
      35.90        1.32
```

✞ Prediction

```
> p=predict(linmod,score_test)
>actuals_preds = data.frame(cbind(actuals=score_test$Final_score,predicts=p))
>actuals_preds
actuals predicts
```

2 69.23 35.89681
11 68.86 62.30306
14 60.00 59.66244
15 60.00 59.66244
16 60.11 67.58432
26 83.99 75.50619
27 73.25 67.58432
32 90.79 67.58432
35 88.38 49.09994
39 51.25 57.02181
41 29.61 49.09994
44 65.13 54.38119
47 78.10 67.58432
50 94.08 72.86557
54 53.33 57.02181
56 51.97 72.86557
57 87.50 75.50619
62 11.11 35.89681
64 86.18 75.50619
70 45.83 54.38119
71 11.11 35.89681
78 64.29 62.30306
79 59.65 54.38119
84 74.34 67.58432
85 5.56 35.89681
88 47.50 59.66244
90 67.50 67.58432
91 71.49 67.58432
93 60.31 62.30306
94 63.78 62.30306
96 73.81 62.30306
99 18.86 72.86557

✞ Print Accuracy

```
> min_max_accuracy=mean(apply(actuals_preds,1,min)/apply(actuals_preds,1,max))
>min_max_accuracy
[1] 0.7806829
>mape=mean(abs((actuals_preds$predicteds-actuals_preds$actuals))/actuals_preds$actuals) >mape
[1] 0.68783456
```

Conclusion: Hence we successfully performed Simple/ Multiple linear regression.