M.SATHISH KUMAR
E0119052

DATA SCIENCE WITH R

CONTINUOS ASSESSMENT - III

1.

Stimulation	no-stimulation
	8
14	6
8	4
7	14
13	
	8

mean = 10.5   4.320

Std = 3.511   18.666

var = 12.33

Count = 4   4

degree of freedom = $[4+4-2] = 6$

Critical value → we should see for 5% percent in Pvalue in table

Calculation:

$$\text{Pvalue} = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

$$= \frac{10.5 - 8}{\sqrt{\frac{12.33}{4} + \frac{18.66}{4}}}$$

$$= \frac{2.5}{\sqrt{3.08 + 4.665}}$$

$$= \frac{2.5}{\sqrt{7.745}}$$

$$= \frac{2.5}{2.78}$$

T value = 0.899

Rcode to show T Test:

```
a = c(14, 8, 7, 13)
b = c(8, 6, 4, 14)
t.test(a, b)
```

Output:

welch to two sample t-test

t = 0.899 df = 9.00 pvalue = .0.4125

Alternative hypothesis ; Difference in mean not equal

to 0

95 percent confidence level

mean x y

10.5 8.0

2. Correlation :

x = c(0, 2, 4, 5, 8, 13, 24, 15, 20)
y = c(12, 15, 16, 14, 22, 24, 28, 30)

cor. test (x, y)

Output :

Pearson product correlation

t = 0.1111   df = 8 , p value = 3.1

Alternative hypothesis true

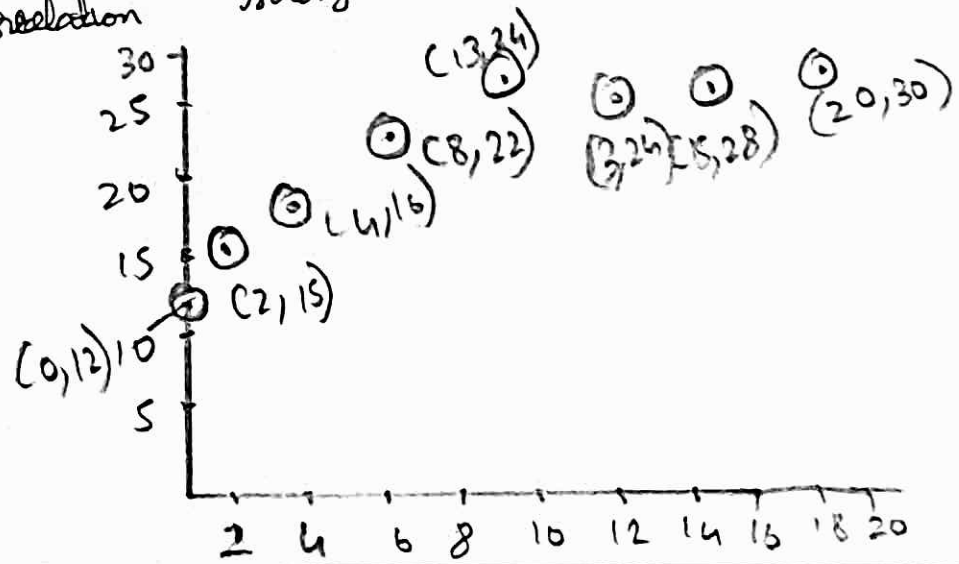
Correlation not equal to 0

95 % confidence   correlation

0.81   0.95   0.81

As data in x increases y also increases. So positive correlation

Correlation strength = strond.



3.

a) **Independent variables :**

It is the variable the experimenter changes or controls and is assumed to have direct effect on dependent variable

Here in Travel. csv Destination , Distance is Independent variables

Dependent variables:

. It is the variable being tested and measured in an experiment

. Dependent on independent values

Here in Travel. csv Airfare is dependent.

b) **Code for Linear regression:**

```
library (caTools)   # Package to split data
df <- read. csv ("Travel. Csv")   # Reading CSV file
set. seed (27)
train_test = sample. split (df, SplitRatio = 0. 7)
# Give output as True False
train = subset (df, split = TRUE)  # Splitted into train, test
test = subset (df, split = FALSE)
```

linear model

model = lm(df $ Airfare ~., data=train)

predict_model = predict (model, test)

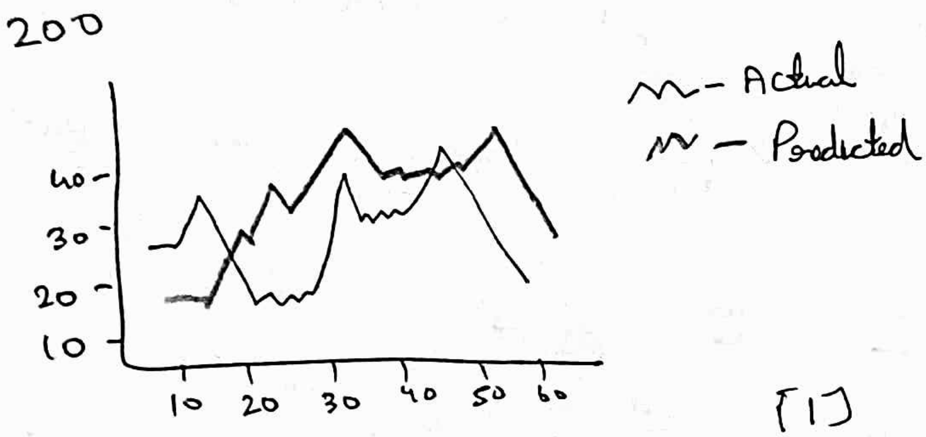
print (~~pr~~ predict_model)
result= data.frame(df $ distance = 150) # Assigning a value
plot (predict_model, type = 'l', col = "green")
pred = predict (model, result)
lines (df $ Airfare, type = 'l', col = "blue")

Plotting and visualizing the predicted and original

Output:

~~Airfare~~	Atlanta	Boston	Chicago	Dallas	......
	176	145	90	281	

miami



[1]
For 150 distance
281 - Airfare

c) Mean Squared Error:

~~import me~~

~~sqrt (mean (predict_model))~~

3c)

```
library ( Metrics)
rmse (df $Airfare , predict_model )
# Gives Root mean squared error
```

Output :

2.813