<u>CS724 Assignment 2</u> (190117)

Step1: Finding the Path Loss Exponent

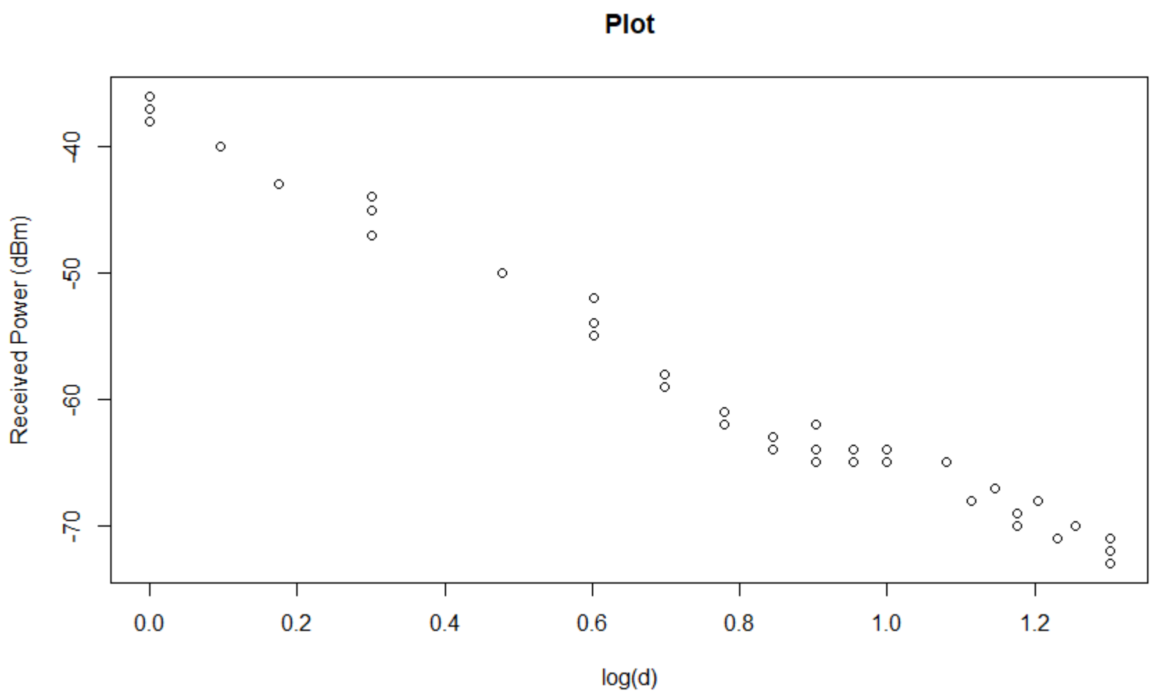
The distance (m) between the AP (transmitter) and the smart phone (receiver) with corresponding Received Power (dBm) values are recorded in '*assignment2.csv*'.

There are total 39 entries for various distances and various orientations of the smart phone. The distance is then converted into its log scale.

Following shows first few entries and a plot of the same using R programming language. The R code is attached with the assignment.

```
> #Reading csv
> ple<-read.csv("assignment2.csv")

> head(ple)
  ReceivedPower_dBm d      LOGd
1               -37 1 0.0000000
2               -45 2 0.3010300
3               -50 3 0.4771213
4               -55 4 0.6020600
5               -59 5 0.6989700
6               -62 6 0.7781512
```



Now we need to find the best fit line and use its slope to find the Path Loss Exponent.

$$P_r(d)\ [dBm] = P_r(d_0)\ [dBm] - 10n\ log_{10}\frac{d}{d_0}$$

For d_0 = 1 m;

$$P_r(d)\ [dBm] = P_r(1)\ [dBm] - 10n\ log_{10}\ d$$

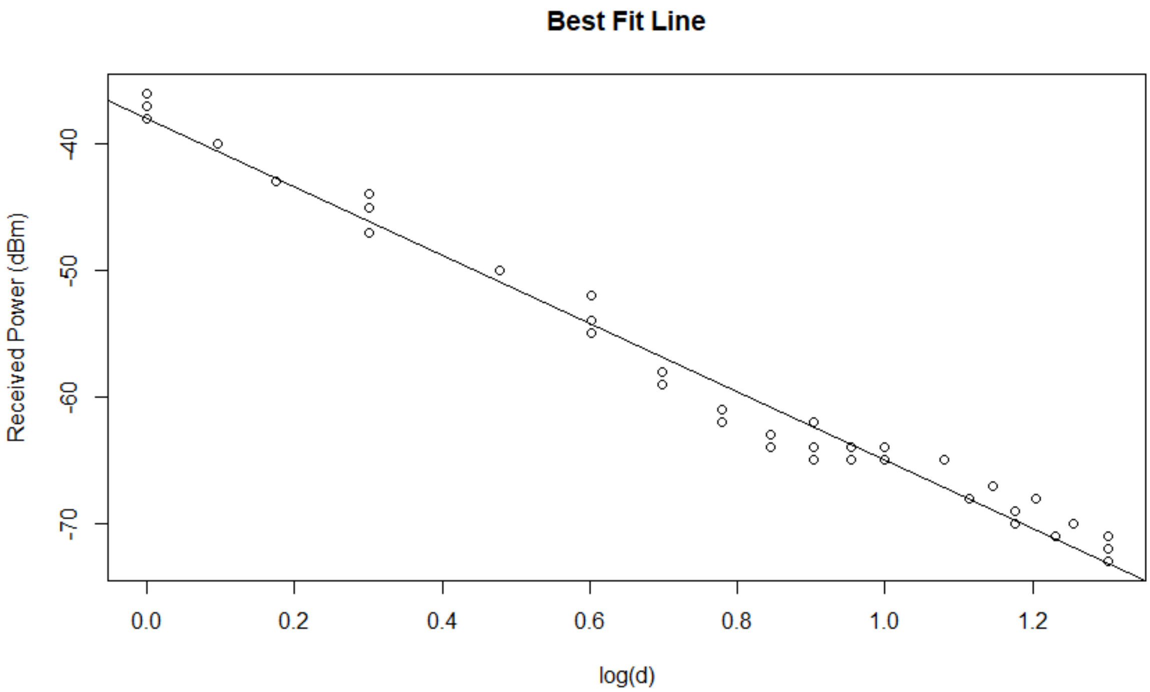
For this, we will develop a linear regression model. A snapshot of the code is following:

```
> #Finding best fit line: Intercept and Coefficient
> model_1 <- lm(ReceivedPower_dBm~LOGd,data=ple)

> model_1

Call:
lm(formula = ReceivedPower_dBm ~ LOGd, data = ple)

Coefficients:
(Intercept)         LOGd
     -38.00       -26.96
```



Hence, we get:

Coefficient of Log of d = -26.96 and Received Power axis intercept = -38.00.

Equivalently,

$$P_r(d)\ [dBm] = -38 - 26.96\ log_{10}\ d$$

So,

$$-10n = -26.96$$

$$\boldsymbol{n = 2.696}$$

Hence, Path Loss Exponent of the room is 2.696

We can get the predicted values of our 39 entries according to the best fit line as shown as below:

```
> #Prediction of our entries according to the best fit line
> predict(model_1,ple)
        1         2         3         4         5         6         7         8         9        10        11        12 
-38.00245 -46.11883 -50.86661 -54.23522 -56.84811 -58.98300 -60.78802 -62.35160 -63.73078 -64.96449 -69.71227 -73.08088 
       13        14        15        16        17        18        19        20        21        22        23        24 
-38.00245 -40.61534 -42.75023 -46.11883 -50.86661 -54.23522 -56.84811 -58.98300 -60.78802 -62.35160 -63.73078 -64.96449 
       25        26        27        28        29        30        31        32        33        34        35        36 
-68.03664 -69.71227 -71.17787 -73.08088 -38.00245 -46.11883 -54.23522 -58.98300 -62.35160 -64.96449 -67.09938 -68.90440 
       37        38        39 
-70.46798 -71.84716 -73.08088 
```

Now, error and variance of the RSSI samples, w.r.t. the best fit line:

```
> #Error w.r.t. the best fit line
> err <- ple$ReceivedPower_dBm - predict(model_1,ple)

> mean(err) #mean of errors
[1] 2.259172e-14

> var(err) #variance of errors
[1] 2.597055
```

Therefore, Variance = 2.597

Step 2: Range Estimation

From the previous step, we get the following equation:

$$P_r(d)\ [dBm] = -38 - 26.96\ log_{10}\ d$$

Path Loss Exponent = 2.696

For d=1 m;

$$P_r(1)\ [dBm] = -38\ dBm$$

$$P_r(1)\ [dB] = -38 - 30 = -68\ dB$$

Now for a given received power value, we can find the distance d as:

$$d = 10 \wedge \left(\frac{-Pr\,(d)\,[dBm] - 38}{26.96}\right)$$

For an entry: P_r = - 53 dBm at a distance d = 4m, we'll find the estimated distance using the above.

$$d = 10 \wedge \left(\frac{53 - 38}{26.96}\right)$$

$$d = 3.60\ m$$

Distance error = Actual distance – estimated distance = 4 – 3.6 = 0.4m.

Now we'll find estimated distance for 5 such new entries. The entries are:

P_r (dBm)	d (m)
-47	2
-53	4
-58	5
-66	10
-69	15

The following code demonstrates the same:

```
> ############STEP 2####################################################################
>
> #5 more entries:
> dist <- c(2,4,5,10,15)

> P_r <- c(-47,-53,-58,-66,-69)

> #estimated distance using the derived equation
> est_dist <- 10^((-P_r-38.00)/26.96)

> est_dist
[1]  2.156889  3.600641  5.518738 10.928880 14.120551

> #error
> err2 <- dist-est_dist

> mean(err2)
[1] -0.06513984
>
```

The average error comes out to be -0.065.