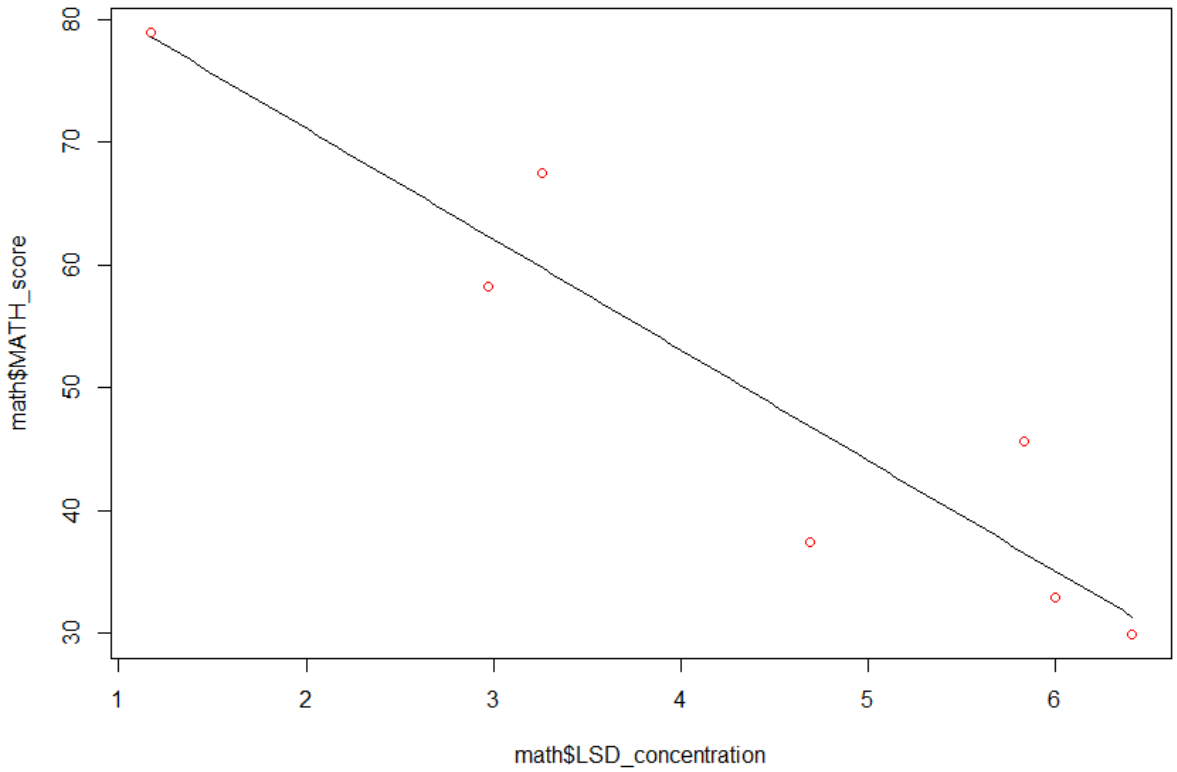
```
> ############# R2 and RMSE Function definition ##########
> r2<-function(y_hat,y) {
+   RSS<-sum((y_hat-(y))^2)
+   TSS<-sum((y-mean(y))^2)
+   return(1-RSS/TSS)}
> rmse=function(y_hat,y)
+ {
+   return(sqrt(mean((y-y_hat)^2)))
+ }
>
> ################ Reading Data files ####################
>
> math <- read.csv("math_scores.csv")
> head(math)
  LSD_concentration MATH_score
1              1.17      78.93
2              2.97      58.20
3              3.26      67.47
4              4.69      37.47
5              5.83      45.65
6              6.00      32.92
> miracle<-read.csv("miracle_food.csv")
> head(miracle)
  Weight_loss pomegranate
1       -0.89           2
2        6.31           2
3      -30.21           3
4       -6.28           7
5       11.38           4
6        1.67           2
>
> ##########################################################
> ############################## Q1 ########################
> ##########################################################
>
> plot(math$MATH_score ~ math$LSD_concentration, col="red")  # scatter plot of the data
> lm.math_coef <- lm(math$MATH_score ~ math$LSD_concentration)    # Least Square fit
> confint(lm.math_coef)
                            2.5 %     97.5 %
(Intercept)             71.00758 107.240169
math$LSD_concentration -12.87325  -5.145685
> summary(lm.math_coef)

Call:
lm(formula = math$MATH_score ~ math$LSD_concentration)

Residuals:
      1       2       3       4       5       6       7
 0.3472 -4.1658  7.7170 -9.3995  9.0513 -2.1471 -1.4032

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)              89.124      7.048  12.646 5.49e-05 ***
math$LSD_concentration   -9.009      1.503  -5.994  0.00185 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.126 on 5 degrees of freedom
Multiple R-squared:  0.8778,   Adjusted R-squared:  0.8534
F-statistic: 35.93 on 1 and 5 DF,  p-value: 0.001854
```

```
> curve(lm.math_coef[[1]][2]*x+lm.math_coef[[1]][1],add=T)       # Linear fit plot
```



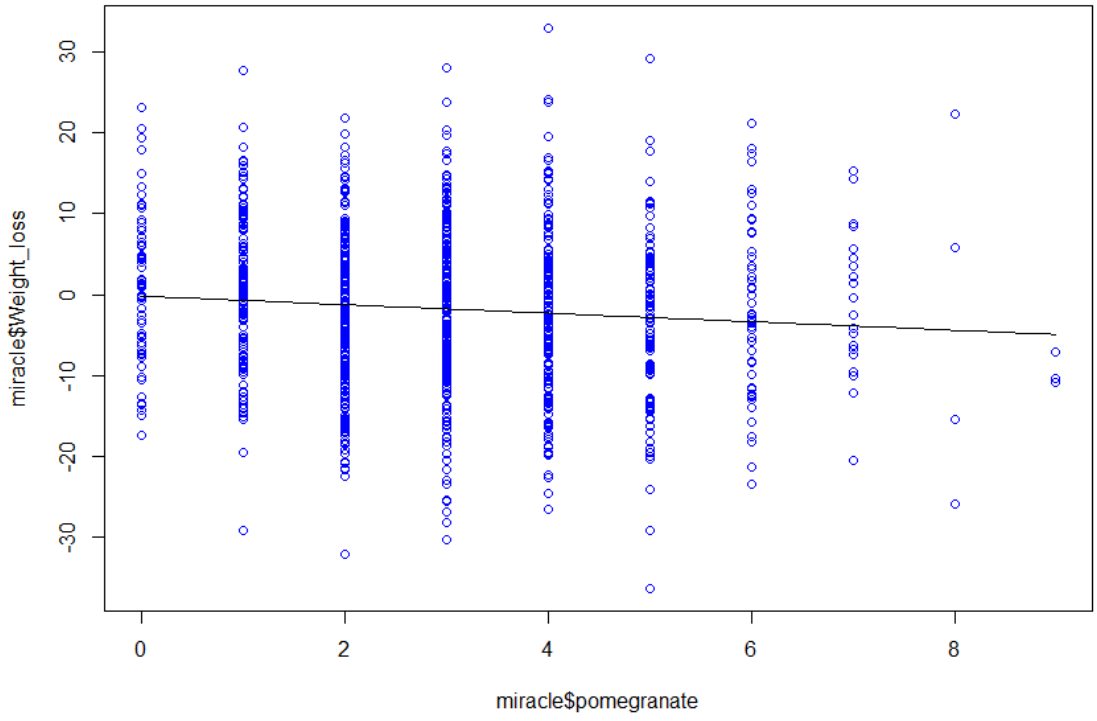

```
>
> # Math Score estimate from Linear Model
> y_hat_math <- lm.math_coef[[1]][2]*math$LSD_concentration+lm.math_coef[[1]][1]
>
> ############### R2 and RMSE values ###############
>
> r2(y_hat_math,math$MATH_score)            # R2 Value
[1] 0.877835
> rmse(y_hat_math,math$MATH_score)          # RMSE Value
[1] 6.022355
>
```

> # **A**.
> # LSD tissue concentration should be less than "LSD_con" for test scores more than 85%

```
> LSD_con <- (85-lm.math_coef[[1]][1])/lm.math_coef[[1]][2]
>
```

> # **B**.
> # LSD concentration does not predict math scores really well because RMSE value of the model is high and also the slope and intercept that fits the model have largeranges (slope between 71 and 107.24 and Intercept between -12.87 and -5.14)
>
> # **C**. Observation vector is not that large to have normal distribution

```
>
> ############################################################
> ############################ Q2 ############################
> ############################################################
>
> plot(miracle$Weight_loss ~ miracle$pomegranate, col="blue") # scatter plot of the data
> lm.miracle_coef <- lm(miracle$Weight_loss ~ miracle$pomegranate)  # Least Square fit
> confint(lm.miracle_coef)
                          2.5 %     97.5 %
(Intercept)         -1.408937  1.0509767
miracle$pomegranate -0.886420 -0.1637906
> summary(lm.miracle_coef)

Call:
lm(formula = miracle$Weight_loss ~ miracle$pomegranate)

Residuals:
    Min      1Q  Median      3Q     Max
-33.435  -6.780  -0.041   6.807  35.169

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.1790     0.6268  -0.286  0.77528
miracle$pomegranate  -0.5251     0.1841  -2.852  0.00444 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.971 on 998 degrees of freedom
Multiple R-squared:  0.008084,  Adjusted R-squared:  0.00709
F-statistic: 8.133 on 1 and 998 DF,  p-value: 0.004435

> curve(lm.miracle_coef[[1]][2]*x+lm.miracle_coef[[1]][1],add=T)    # Linear fit plot
```



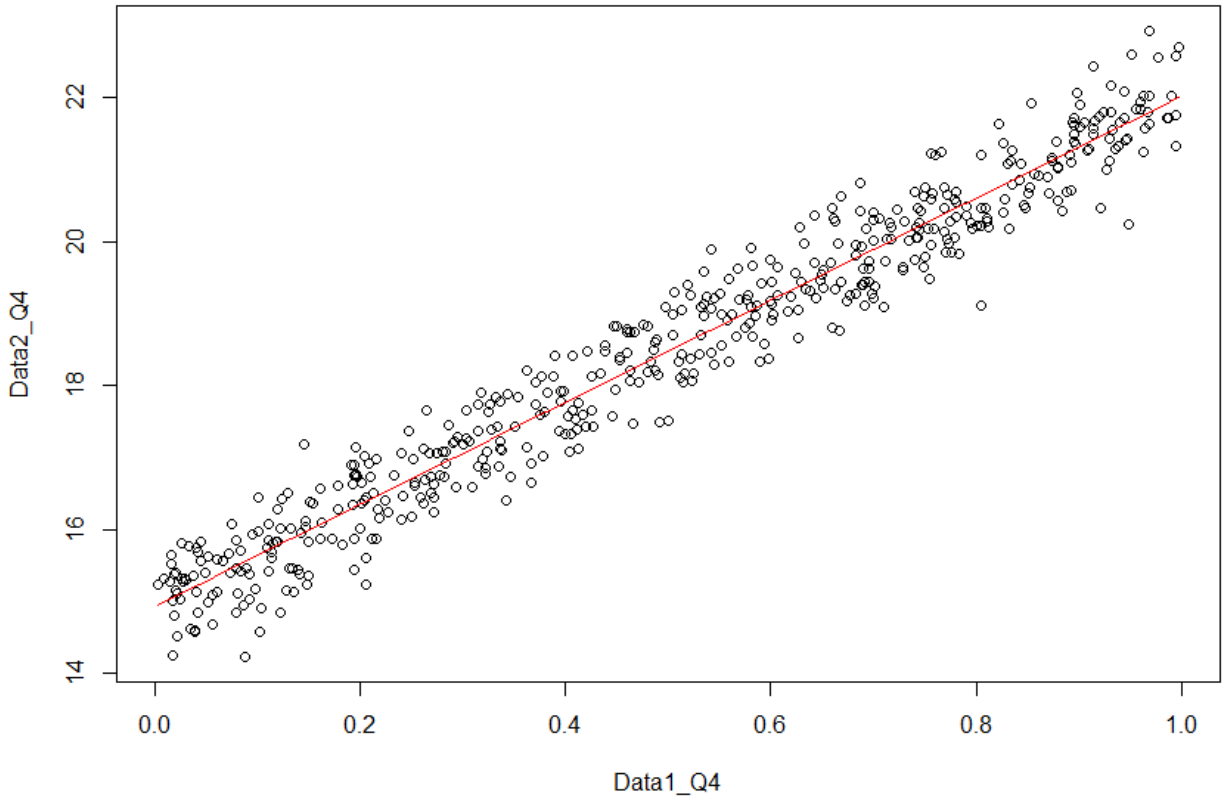
```
> # Weight loss estimate from Linear Model
> y_hat_miracle <- lm.miracle_coef[[1]][2]*miracle$pomegranate+lm.miracle_coef[[1]][1]

> ############### R2 and RMSE values ###############
>
> r2(y_hat_miracle,miracle$weight_loss)              # R2 Value
[1] 0.008083812
> rmse(y_hat_miracle,miracle$weight_loss)            # RMSE Value
[1] 9.961044
>
```

> # I disagree, although the P-value is about 0.0044, R2 value is so low which linear model with negative slope is not appropriate for observations.

```
> ############################################################
> ############################ Q3 ############################
> ############################################################
>
> ######################## MAE Function ######################
> MAE<-function(y_hat,y) {
+   return(sum(abs(y_hat-y))/length(y))
+ }
> ################ R2, RMSE and MAE values for Math ###############
> r2(y_hat_math,math$MATH_score)              # R2 Value
[1] 0.877835
> rmse(y_hat_math,math$MATH_score)            # RMSE Value
[1] 6.022355
> MAE(y_hat_math,math$MATH_score)             # Mean Absolute Error
[1] 4.890144
>
> ############ R2, RMSE and MAE values for Weight Loss ############
> r2(y_hat_miracle,miracle$Weight_loss)       # R2 Value
[1] 0.008083812
> rmse(y_hat_miracle,miracle$Weight_loss)     # RMSE Value
[1] 9.961044
> MAE(y_hat_miracle,miracle$Weight_loss)      # Mean Absolute Error
[1] 7.981461
>
```

> # In the first question though R2 value seems reasonable, RMSE and MAE values are large which means the model does not show a solid relation between math scores and LSD tissue.
> # In question 2, low R2 and high RMSE and MAE values all represent that the model does not fit the observations very well

```
>
> ############################################################
> ############################ Q4 ############################
> ############################################################
> Data1_Q4 <- runif(500)
> slope <- 7
> intercept <- 15
> Data2_Q4<-rnorm(500,mean=intercept+slope*Data1_Q4,0.45) # Normally distributed random
variable
> plot(Data2_Q4 ~ Data1_Q4)                                  # scatter plot of the data
> lm_coef <- lm(Data2_Q4 ~ Data1_Q4)                         # Least Square fit
> curve(lm_coef[[1]][2]*x+lm_coef[[1]][1],add=T,col="red")   # Linear fit plot
```

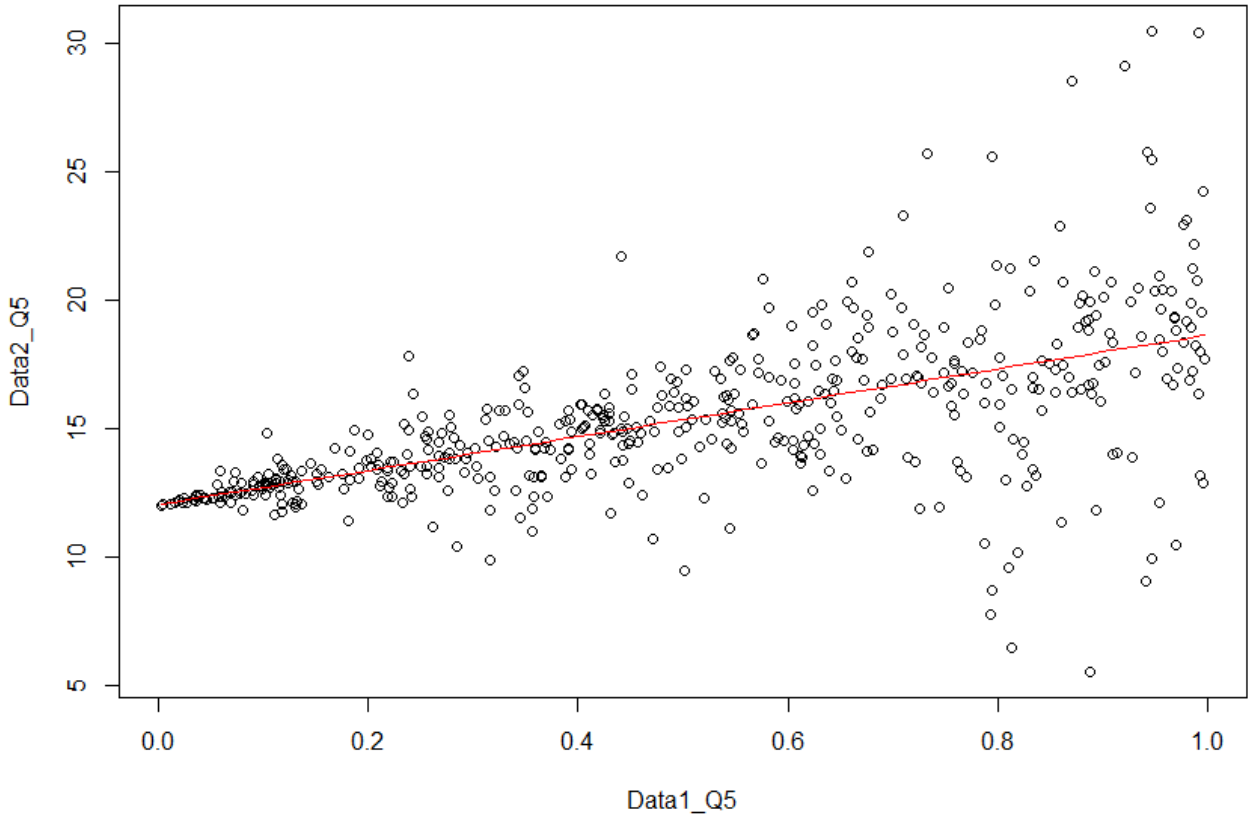


C.

As you see the estimated slope and intercept, it seems that they are close to the values we came up with.

```
> confint(lm_coef)
                2.5 %    97.5 %
(Intercept) 14.899280 15.059936
Data1_Q4     6.895731  7.168381
>
```

```
> ############################################################
> ############################ Q5 ############################
> ############################################################
> Data1_Q5 <- runif(500)
> slope <- 7
> intercept <- 12
> step_value <- abs(rep(c(1:10)*0.7,times=50))
> sigma <- step_value[order(step_value)]*Data1_Q5            # Incremental Variance value
> Data2_Q5 <- rnorm(500,mean=intercept+slope*Data1_Q5,sigma) # Normally distributed random
variable
> plot(Data2_Q5 ~ Data1_Q5)                                  # scatter plot of the data
> lm_coef <- lm(Data2_Q5 ~ Data1_Q5)                         # Least Square fit
> curve(lm_coef[[1]][2]*x+lm_coef[[1]][1],add=T,col="red")   # Linear fit plot
```




```
> confint(lm_coef)
                2.5 %    97.5 %
(Intercept) 11.637667 12.502981
Data1_Q5     6.132296  7.644942
```