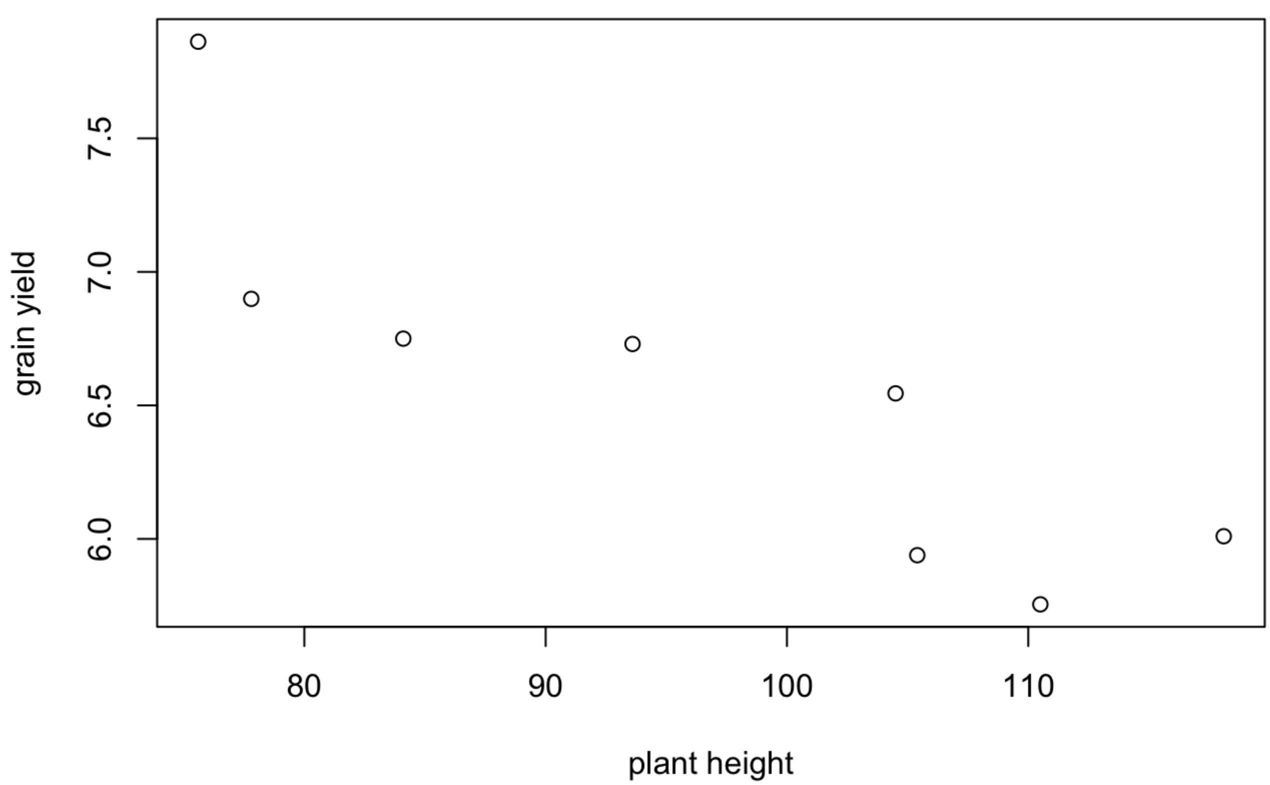
Hmw_06

Evelin Reyes

11/8/2021

Q1

```
d = data.frame(x = c(110.5, 105.4, 118.1, 104.5, 93.6, 84.1, 77.8, 75.6),
y = c(5.755, 5.939, 6.010, 6.545, 6.730, 6.750, 6.899, 7.862))
plot(d$x, d$y, xlab = "plant height", ylab = "grain yield")
```



Answer a

```
cor(d$x, d$y)
```

```
## [1] -0.868707
```

```
fit_d = lm(y ~ x, data = d)
summary(fit_d)
```

```
##
## Call:
## lm(formula = y ~ x, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34626 -0.27605 -0.09448  0.27023  0.53495
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.137455   0.842265  12.036    2e-05 ***
## x           -0.037175   0.008653  -4.296  0.00512 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3624 on 6 degrees of freedom
## Multiple R-squared:  0.7547, Adjusted R-squared:  0.7138
## F-statistic: 18.46 on 1 and 6 DF,  p-value: 0.005116
```

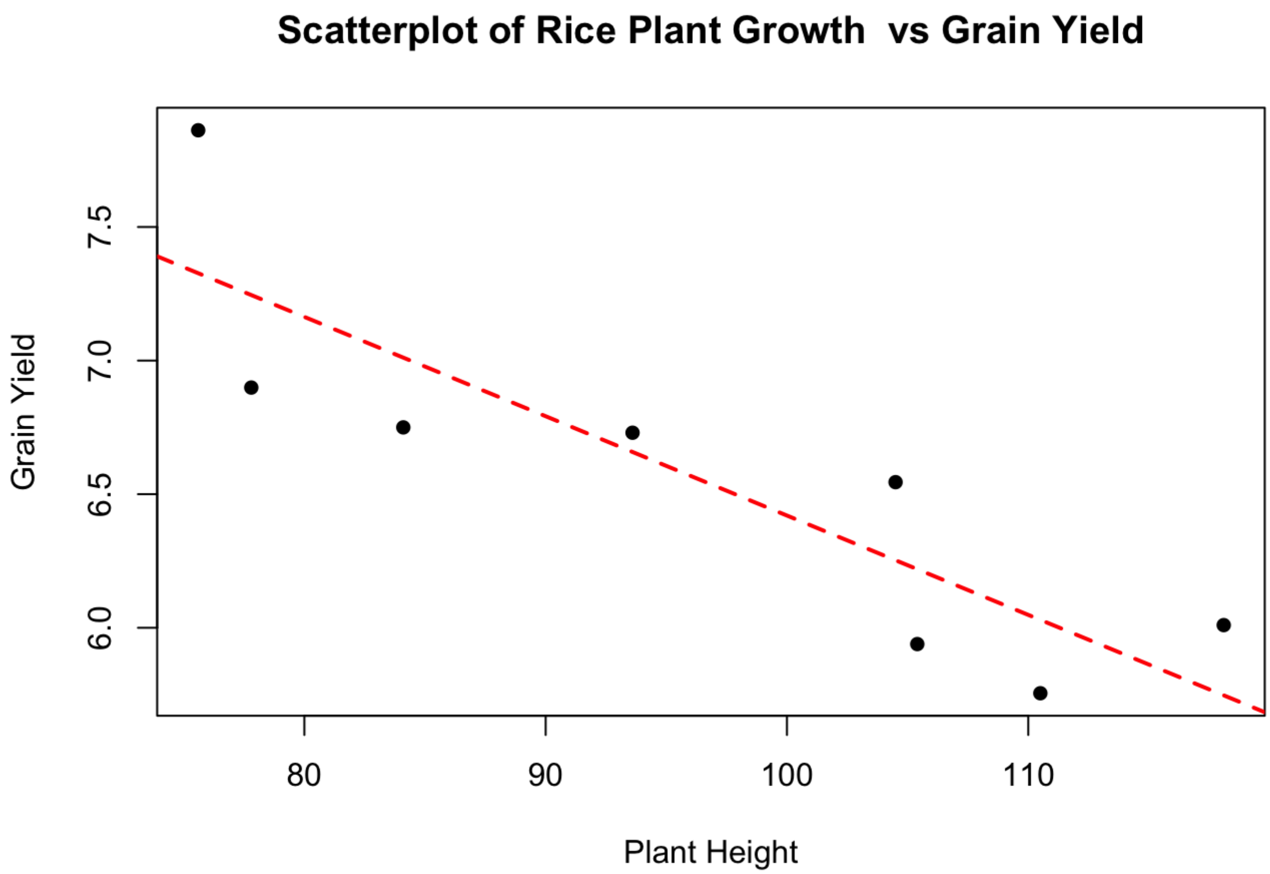
```
#beta 0
coef(fit_d)[1]
```

```
## (Intercept)
##    10.13746
```

```
# beta 1  Units??
coef(fit_d)["x"]
```

```
##           x
## -0.03717469
```

```
plot(d$x, d$y, pch=16,
     xlab = "Plant Height",
     ylab = "Grain Yield",
     main = "Scatterplot of Rice Plant Growth  vs Grain Yield") +
abline(coefficients(fit_d), lwd = 2, lty = 2,
       col = "red")
```




```
## integer(0)
```

```
## the estimate tells us that the grain yield will decrease as the plant gets taller. Th
e regression line indicates that yield will likely fall below 6.0 once plant height is m
ore than about 105
```

Answer b

```
anova(fit_d)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## x          1 2.42357 2.42357  18.455 0.005116 **
## Residuals  6 0.78794 0.13132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fit_d)
```

```
##
## Call:
## lm(formula = y ~ x, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34626 -0.27605 -0.09448  0.27023  0.53495
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.137455   0.842265  12.036    2e-05 ***
## x           -0.037175   0.008653  -4.296  0.00512 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3624 on 6 degrees of freedom
## Multiple R-squared:  0.7547, Adjusted R-squared:  0.7138
## F-statistic: 18.46 on 1 and 6 DF,  p-value: 0.005116
```

```
## Both the F test and T test gave a p-value of less than 5%, allowing us to reject a nu
ll hypothesis of the slope value = 0
```

Answer C

```
qt(0.05/2, 8-2)
```

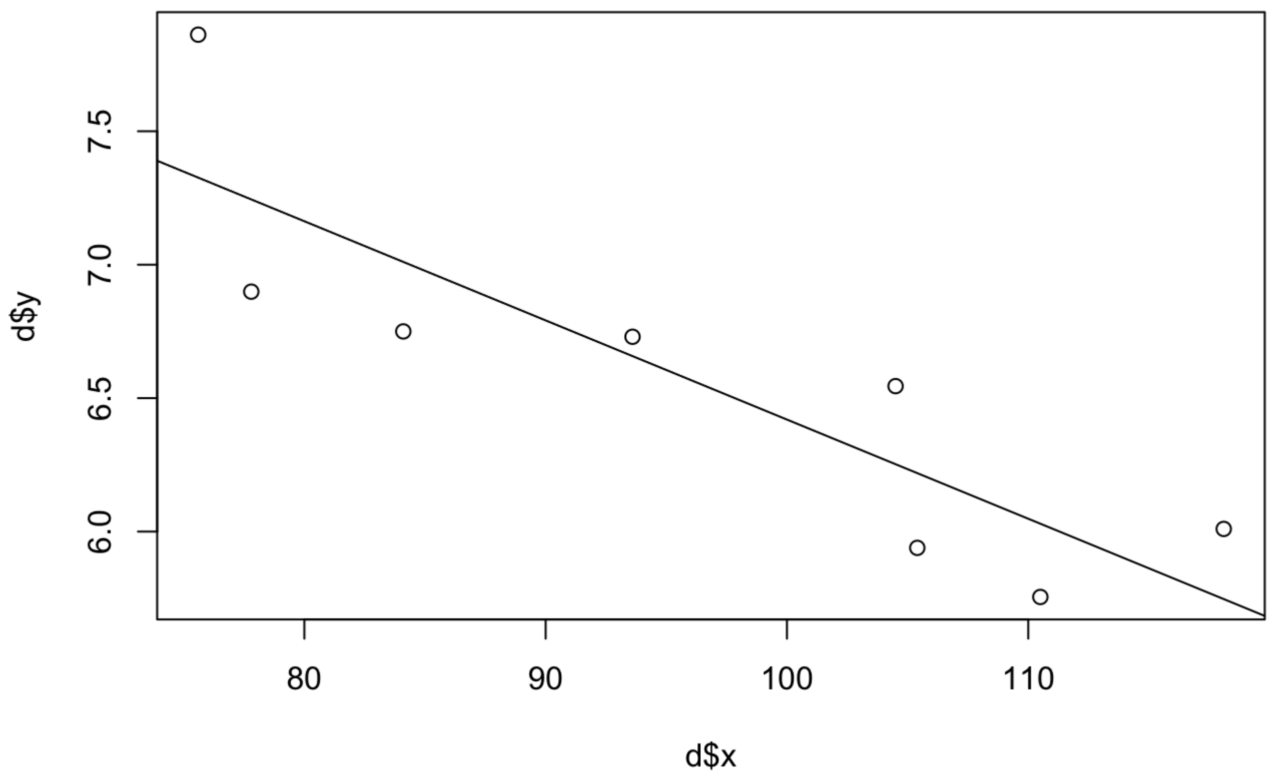
```
## [1] -2.446912
```

```
# 95% of the time that the confidence interval is calculate it will overlap with the tru
e value of the slope.
confint(fit_d)
```

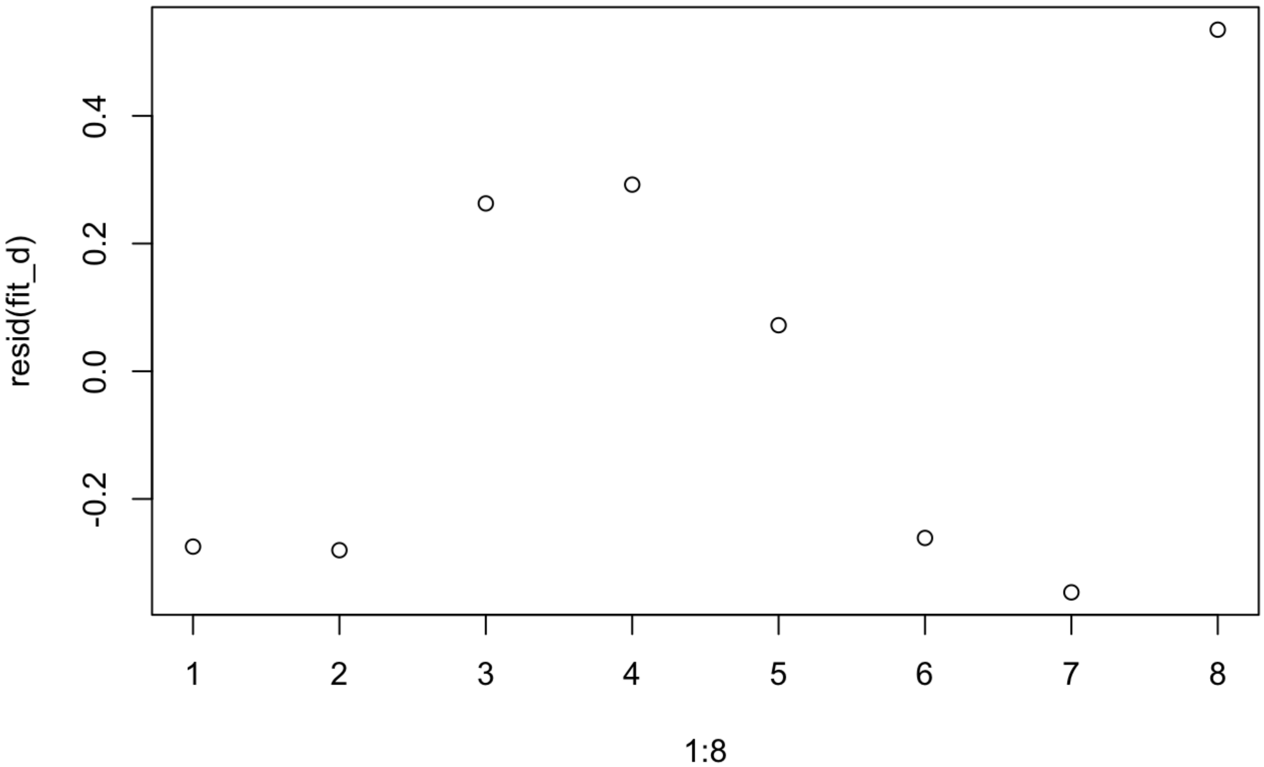
```
##                   2.5 %      97.5 %
## (Intercept)  8.07650745 12.19840320
## x           -0.05834895 -0.01600043
```

Answer d

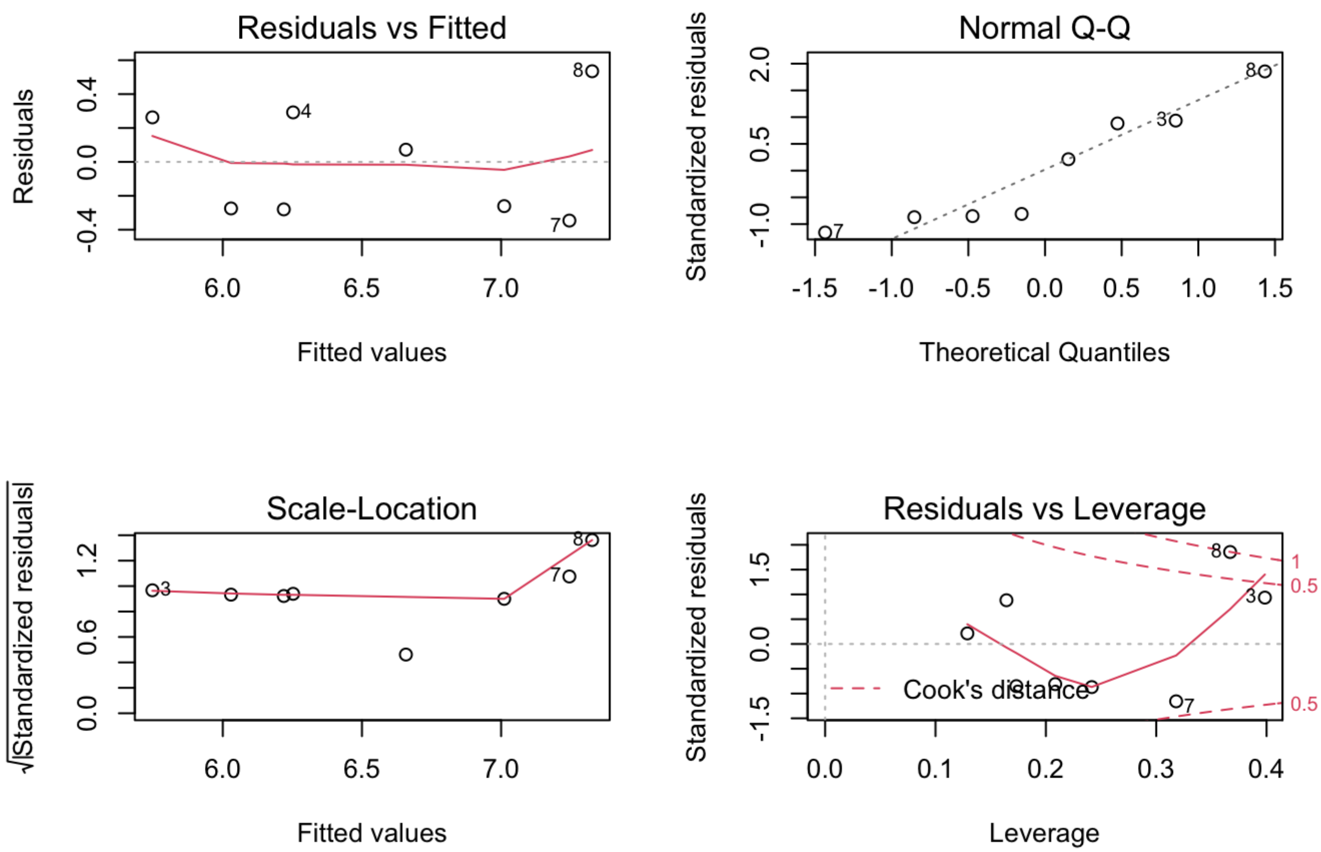
```
{plot(d$x,d$y)
abline(fit_d)}
```




```
{y=plot(resid(fit_d), x=1:8)
abline(fit_d)}
```




```
### resid(fit_d) gives me the actual value of the residuals
### abline(fit_d) plots the slope
{par(mfrow=c(2,2)) ## plotting the residuals vs fitted lines.
plot(fit_d)}
```



Answer e

```
SSE<-sum((fitted(fit_d) - d$y)^2)

ev<-SSE/(8-2)
sum(fit_d$residuals**2)
```

```
## [1] 0.7879367
```

```
ev1<-anova(fit_d)
ev1<-ev1$`Mean Sq`[2]
```

Answer f

```
est<-predict(fit_d, newdata = data.frame(x = 100), interval = "confidence")
####int[1] its how much the 100 things is gonna yield, int[2] is the lower bound of the
 ci and int[3] is the upper bound
```

Answer g

```
pred<-predict(fit_d, newdata = data.frame(x = 100), interval = "prediction")
####int[1] its how much the 100 things is gonna yield, int[2] is the lower bound of the
 ci and int[3] is the upper bound
```

Answer h

```
sumar<-summary(fit_d)
sumar$r.squared
```

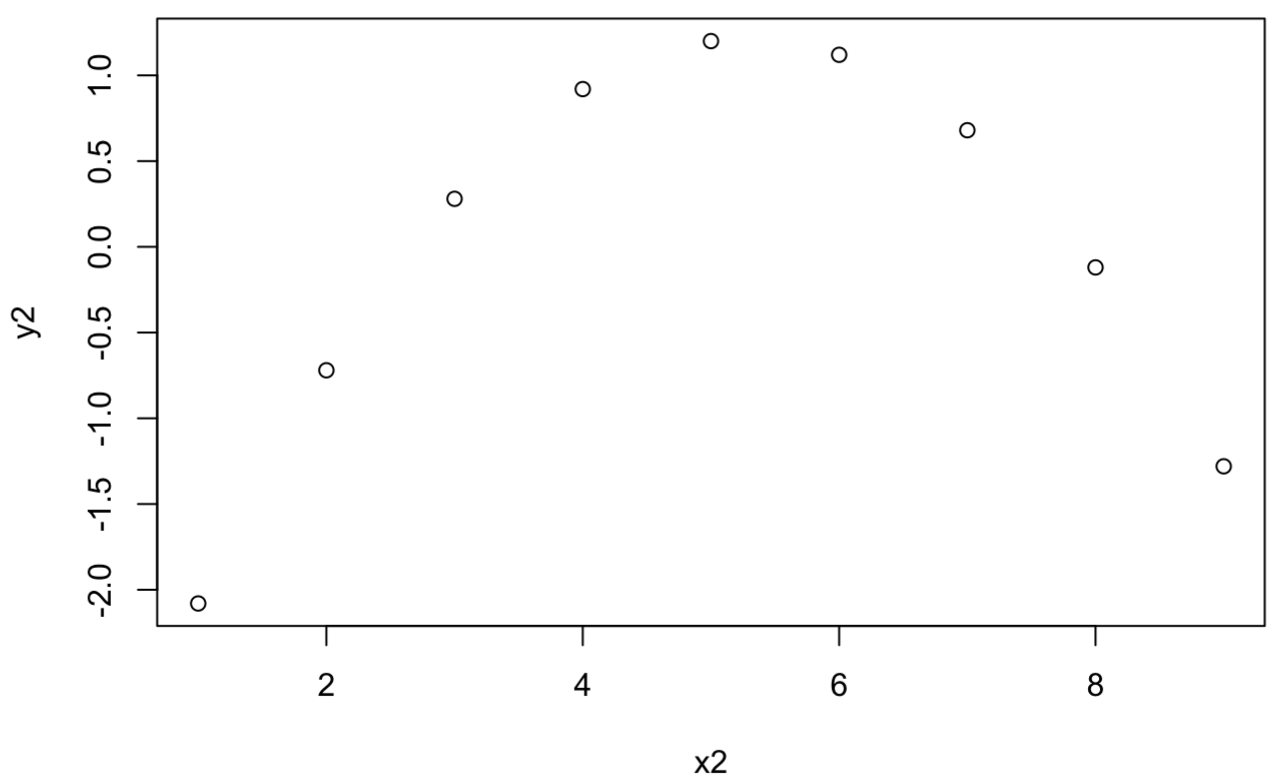
```
## [1] 0.7546518
```

```
##### This means how well the model fits the data
```

Q2

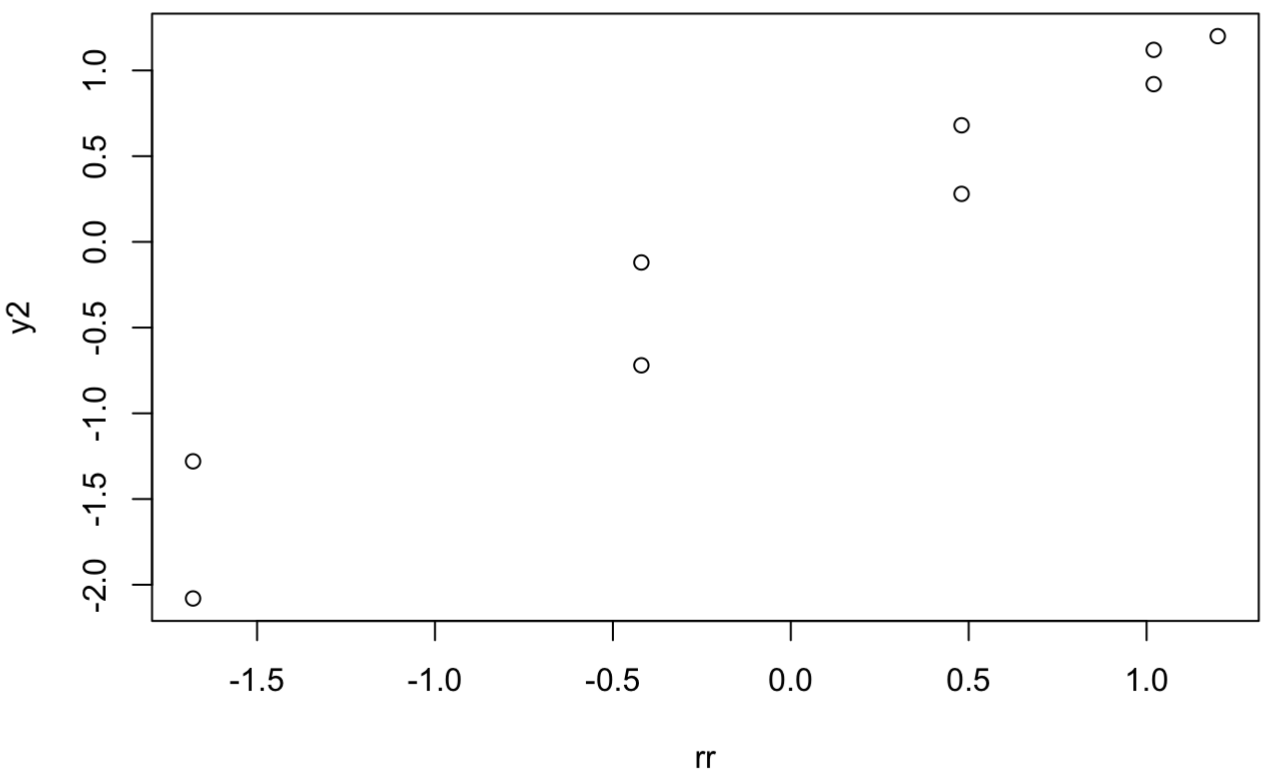
2a answer

```
x2 = c(1, 2, 3, 4, 5, 6, 7, 8, 9)
y2 = c(-2.08, -0.72, 0.28, 0.92, 1.20, 1.12, 0.68, -0.12, -1.28)
plot(x = x2, y = y2)
```



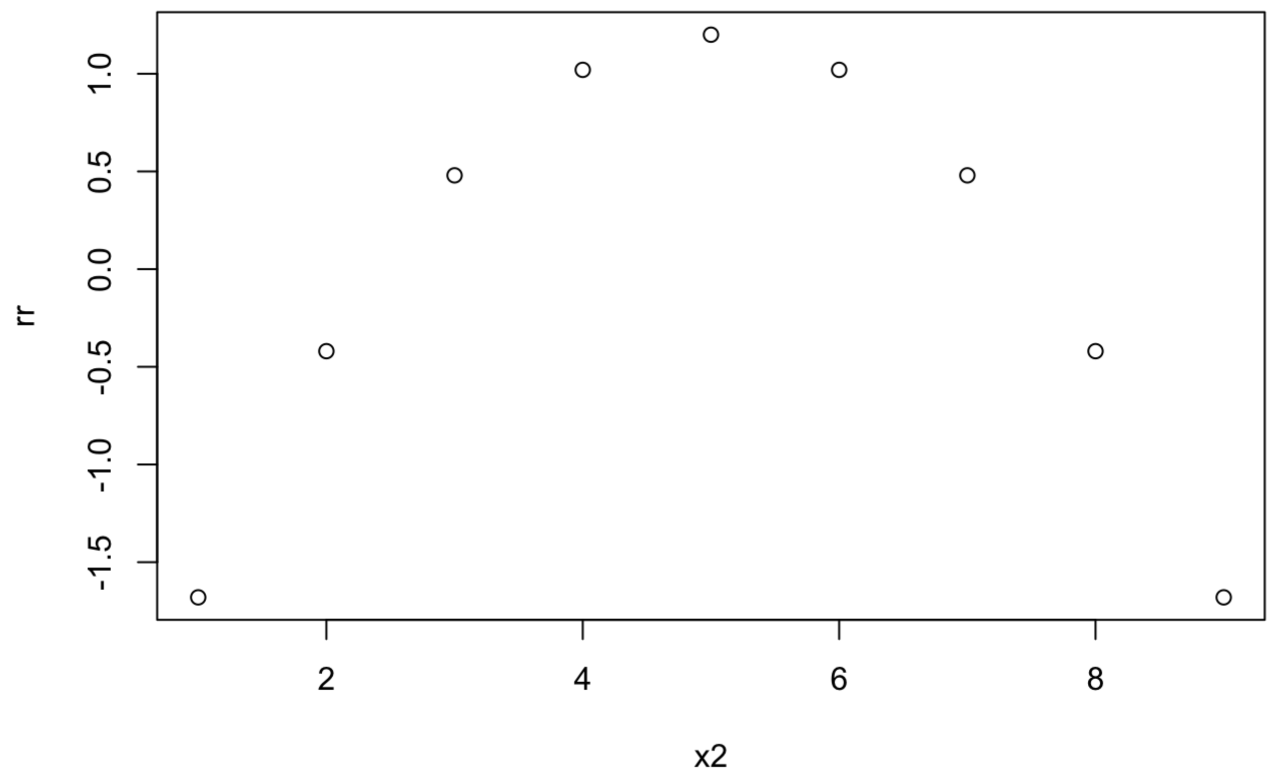
2b answer

```
modelo<-lm(y2~x2)
rr<-resid(modelo)
plot(x = rr, y = y2)
```



2c answer

```
plot(x = x2, y = rr)
```

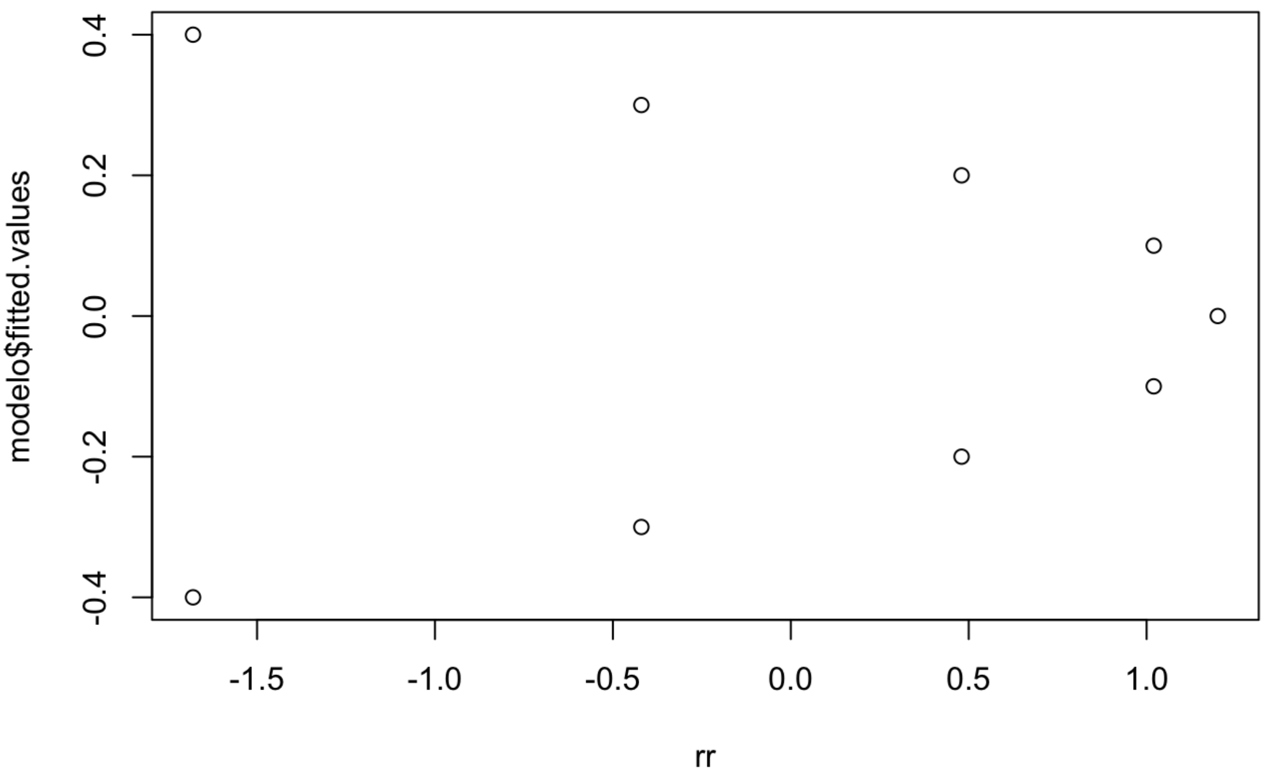


2d answer

```
modelo$fitted.values
```

```
##             1             2             3             4             5 
## -4.000000e-01 -3.000000e-01 -2.000000e-01 -1.000000e-01 -2.220446e-16 
##             6             7             8             9 
##  1.000000e-01  2.000000e-01  3.000000e-01  4.000000e-01
```

```
plot(x = rr, y = modelo$fitted.values)
```



2e answer

Yes, there is a meaningfull difference between plot c and d. In ### plot d, rr value maximises when fitted value is at zero. In plot ### c, rr value does not respond this way, instead maximizing aroudn five.

Plot b is a better indication of lack of fit, because the points are close fit to a line of regression.