STATS FALL 2021

Problem Set 02

Kathryn Glen

17331061

10/13/2021

Copying set up from template given in first Problem Set

```
# remove objects
rm(list=ls())

# set working directory
#setwd("~/Documents/GitHub/QTM200Spring2021/problem_sets/PS1")
setwd("/Users/Kate/Desktop/Hacker/StatsI_Fall2021-main/problemSets/PS02")
```

QUESTION ONE

Q1(a) Calculate the $\chi 2$ test statistic by hand (even better if you can do "by hand" in R).

Calculate Chi sq satistic by hand in R

Null hypothesis = upper class bribes = lower class bribes Alternative hypothesis = upper class bribes not equal lower class bribes

Overall number of observations = 42 Need to determine Fe = row total/grand total multiplied by column total Upper class Fe:

```
((14+6+7) * (6+7) / (42))
```

```
## [1] 8.357143
```

```
UFe=8.36
```

Upper class Fexpected=8.36

Lower class Fe:

```
((7+7+1) *  (6+7) /(42))
```

```
## [1] 4.642857
```

Lower class (Fexpected= 4.64) Lower class (F observed=7)

Upper class (Fobserved= 6) Lower class (Fobserved = 7)

```
UFe=8.36
UFo=6
LFe=4.64
LFo=7
```

Chi sq stat = sum of (fo-fe)^2 / fe Check upper class chi and lower class chi, then add together

Chi Upper class:

```
(UFo-UFe)^2/UFe
```

```
## [1] 0.6662201
```

ChiU= 0.6662201

Chi Lower class:

```
(LFo-LFe)^2/LFe
```

```
## [1] 1.200345
```

Sum= 1.200345

To get the chi statistic

```
0.6662201 +1.200345
```

```
## [1] 1.866565
```

ANSWER: 1.866565

Q1(b) Now calculate the p-value from the test statistic you just created (in R).2 What do you conclude if α = .1?

Calculate p-value, alpha = .1 pchisq df = (rows-1)(columns-1), 2 rows, 3 columns

```
degreesf= (3-1)*(2-1)
degreesf
```

```
## [1] 2
```

Calculate the p-value in R using the method from the slides

```
pval=pchisq(1.866565, df=2, lower.tail=FALSE)
pval

## [1] 0.3932607
```

P-value = 0.3932607

Alpha value = .1

Since the p-value is considerably higher than the alpha value, we cannot reject the null hypothesis.

This means that in this study there was no real difference between the number of bribes offered to upper class and lower class participants by police at traffic stops. This study suggests that officers were not any more or less likely to solicit bribes depending on class.

Q1(c) Calculate the standardized residuals for each cell and put them in the table below.

Need to create the table in R before finding the standardised residuals of each cell

Below blueprint found on statology to create a table in R:

tab <- matrix(c(7, 5, 14, 19, 3, 2, 17, 6, 12), ncol=3, byrow=TRUE) colnames(tab) <- c('colName1','colName2','colName3') rownames(tab) <- c('rowName1','rowName2','rowName3') tab <- as.table(tab)

```
table <- matrix(c(14,6,7,7,7,1), ncol=3, byrow=TRUE)
colnames(table) <- c('Not Stopped', 'Bribe requested', 'Stopped/Given warning
')
rownames(table) <- c('Upper Class', 'Lower Class')
table <- as.table(table)
table

##             Not Stopped Bribe requested Stopped/Given warning
## Upper Class          14               6                     7
## Lower Class           7               7                     1
```

Recreated the table from the assignment in R.

I struggled to get standardized residuals for quite a while:

The ***\\ indicates where I managed to get it done.

First attempt:

To calculate the standardised residuals I need the Fe and the Fo of every cell. Fe = row total/grand total * column total

```
UCTotal= 14+6+7
LCTotal=7+7+1
GrandTot= 42

#Upper Class Fes
FeUNS= UCTotal/GrandTot * (14+7) #13.5
FeUBR= UCTotal/GrandTot * (6+7) #8.357
FeUSG= UCTotal/GrandTot * (7+1) #5.143

#Upper Class Fos
FoUNS= 14
FoUBR= 6
FoUSG= 7

#Lower Class Fes
FeLNS= LCTotal/GrandTot * (14+7) #7.5
FeLBR= LCTotal/GrandTot * (6+7) #4.643
FeLSG= LCTotal/GrandTot * (1+7) #2.857

#Lower Class Fos
FoLNS= 7
FoLBR= 7
FoLSG= 1
```

I have calculated the Fe and Fo of every cell using the above code. Now I plug that information into the formula for standardised residuals given in the lecture.

z = fo-fe/se

The se is the square root of fe(1-row.prop.)(1-column.prop.)

Based on the information in the slides the: Row.prop. = row total / grand total Column prop = column total / grand total

Now I need to calculate Z for every cell in the table

```
#Upper class:

#ZUNS
(FoUNS- FeUNS)

## [1] 0.5

#0.5

#ZUNS=.5/sqrt(13.5*((1-UCTotal/GrandTot)*(1-(14+7)))
```

I have tried a lot to get R to do this sum and even when it accepts the code it won't show me the answer to the sum, so I have to do this sum by hand and input the details.

Is there a simpler way to do this in R that you could show us in class?

The answer keeps coming up as error on calculator or undefined, the maths appears to be tripping up at the square root, so I am going to try find a function in r that willl make this work.

Second Attempt: Create regression model, find standardised residuals through thtat using instructions found on statology.org

Regression model using table from above.

This did not work: model<- lm(Upper Class ~ Lower Class, data=table) summary(model)

Third Attempt:

Trying again by creating the table as a dataset.

```
bribes <- data.frame(x=c(14,6,7),
                     y=c(7,7,1))
bribes
```

```
##    x y
## 1 14 7
## 2  6 7
## 3  7 1
```

Fit regression model

```
model<- lm(y~x, data=bribes)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = bribes)
##
## Residuals:
##       1       2       3
##  0.4211  2.9474 -3.3684
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1579     7.0581   0.306    0.811
## x             0.3158     0.7293   0.433    0.740
##
## Residual standard error: 4.496 on 1 degrees of freedom
## Multiple R-squared:  0.1579, Adjusted R-squared:  -0.6842
## F-statistic: 0.1875 on 1 and 1 DF,  p-value: 0.7399
```

Calculate the standardised residuals using rstandard()

```
stand.res<- rstandard(model)
stand.res
```

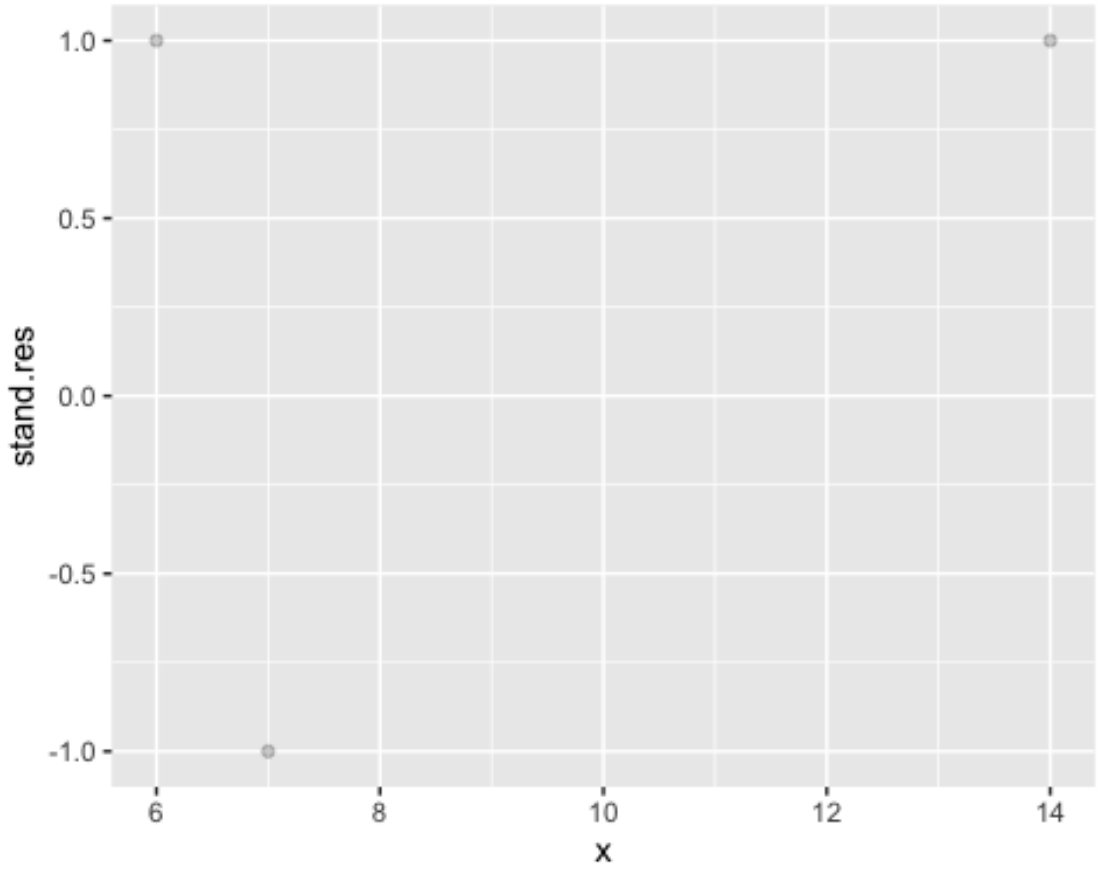
```
##  1  2  3
##  1  1 -1
```

This is odd, as it only gives my three standardised residuals when there should be six. I will try and plot it out to visualise the problem.

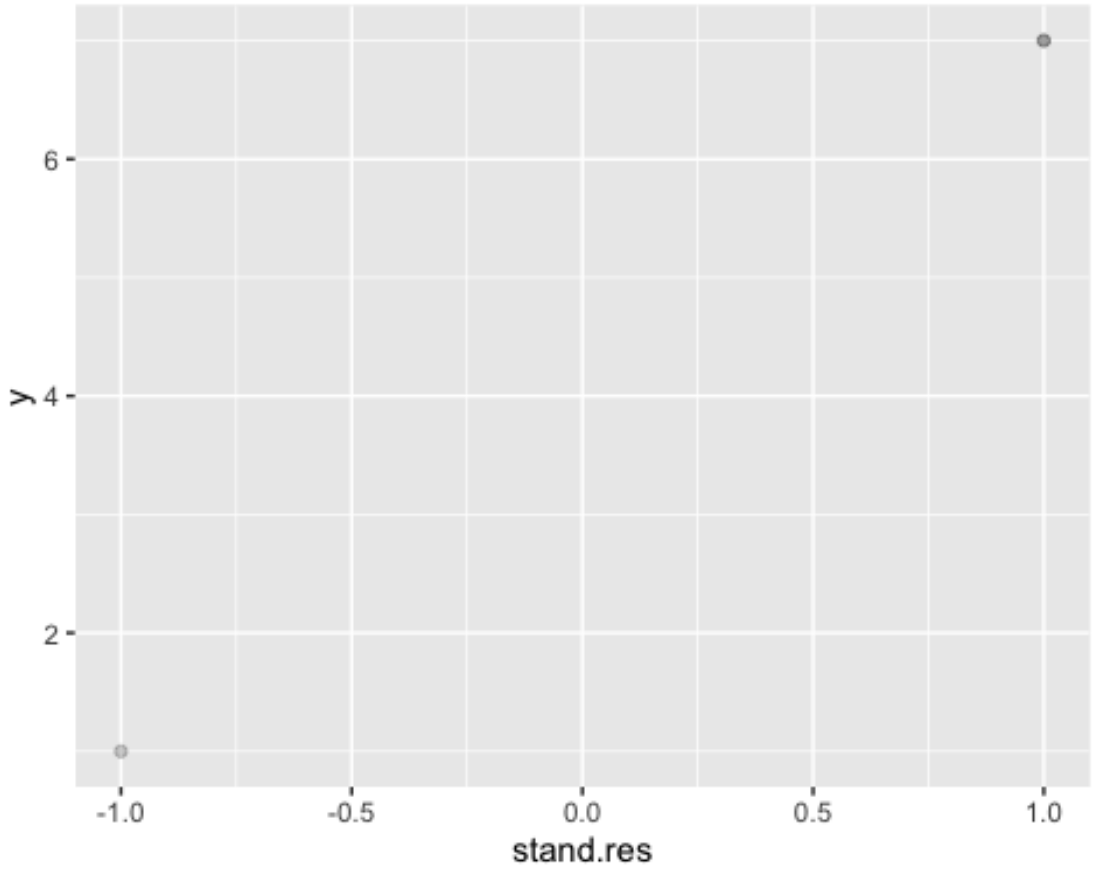
```
Final_stand.res<- cbind(bribes, stand.res)
Final_stand.res
```

```
##    x y stand.res
## 1 14 7         1
## 2  6 7         1
## 3  7 1        -1
```

```
library(ggplot2)
ggplot(aes(x, stand.res), data = Final_stand.res) +
  geom_point(alpha = 0.2)
```



```
ggplot(aes(stand.res, y), data = Final_stand.res) +
  geom_point(alpha = 0.2)
```



plotting

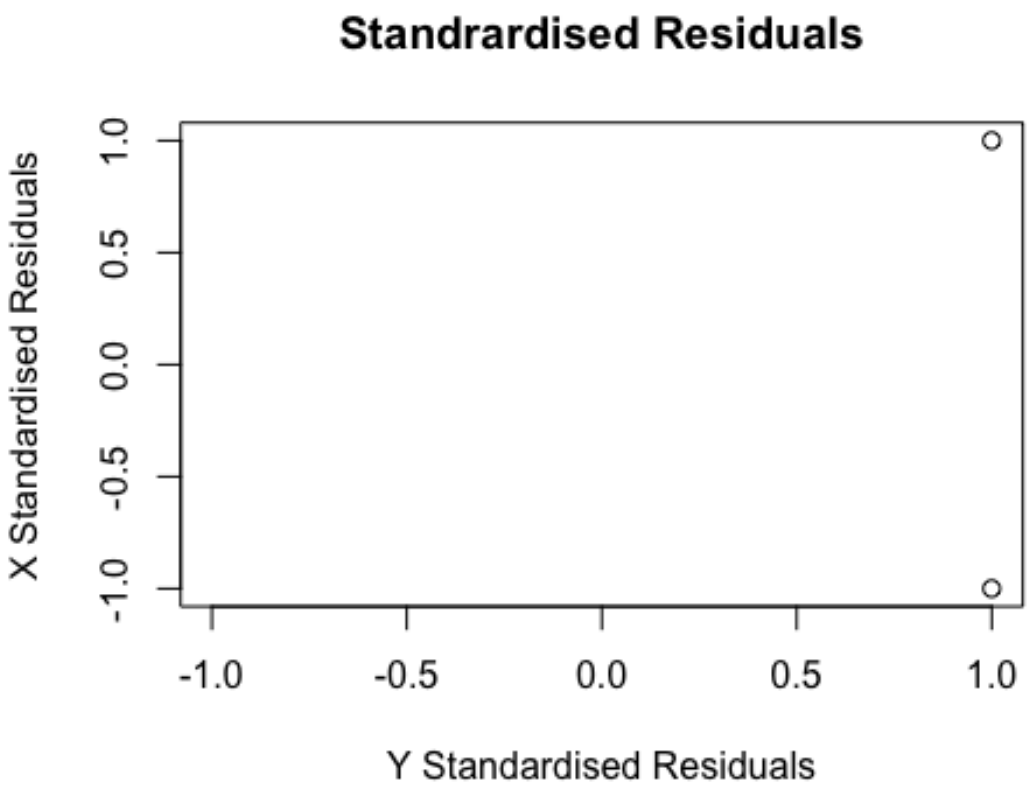
Plotting out the problem did not get me any further

Fourth Attempt:

The below code is from r-tutor.com

```
bribes.lm=lm(x~y, data=bribes)
bribes.stdres=rstandard(bribes.lm)
twobribes.lm=lm(y~x, data=bribes)
twobribes.stdres=rstandard(twobribes.lm)

plot(twobribes.stdres, bribes.stdres,
     ylab="X Standardised Residuals",
     xlab= "Y Standardised Residuals",
     main= "Standrardised Residuals")
```



This also did not make anything any clearer.

Fifth Attempt:

Trying again but with two tables

```
UpperClass <- data.frame(x=c("NotStopped","Bribes","Stopped"),
                         y=c(14,6,7))
UpperClass
```

```
##            x  y
## 1 NotStopped 14
## 2     Bribes  6
## 3    Stopped  7
```

This created a table for Upper Class, now to find the Standardised Residuals of these cells

```
model.upper<- lm(y~x, data=UpperClass)
summary(model.upper)
```

```
##
## Call:
## lm(formula = y ~ x, data = UpperClass)
```

```
##
## Residuals:
## ALL 3 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6        NaN     NaN      NaN
## xNotStopped        8        NaN     NaN      NaN
## xStopped           1        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 2 and 0 DF,  p-value: NA
```

```
upper.stand.res <- rstandard(model.upper)
upper.stand.res
```

```
##   1   2   3
## NaN NaN NaN
```

The only outcome is NaN.

Sixth Attempt:

I will try again with the lower class table, but this is not working and I don't know why, and I am running out of ideas.

```
LowerClass <- data.frame(x=c("NotStopped","Bribes","Stopped"),
                         y=c(7,7,1))
LowerClass
```

```
##           x y
## 1 NotStopped 7
## 2     Bribes 7
## 3    Stopped 1
```

```
model.lower<- lm(y~x, data=LowerClass)
summary(model.lower)
```

```
##
## Call:
## lm(formula = y ~ x, data = LowerClass)
##
## Residuals:
## ALL 3 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.000e+00        NaN     NaN      NaN
## xNotStopped  1.632e-15        NaN     NaN      NaN
```

```
## xStopped    -6.000e+00        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 2 and 0 DF,  p-value: NA
```

```
lower.stand.res <- rstandard(model.lower)
lower.stand.res
```

```
##   1   2   3
## NaN NaN NaN
```

Again all I get is NaN.

***\\

Final attempt: using the formula for standardised residuals given by statisticshowto.com
Standardized residuals= (observed count - expected count) / sqrt(expected count)

```
ResUNS= (FoUNS-FeUNS)/sqrt(FeUNS) #0.136
ResUBR= (FoUBR-FeUBR)/sqrt(FeUBR) #-0.815
ResUSG= (FoUSG-FeUSG)/sqrt(FeUSG) #0.819

ResLNS= (FoLNS-FeLNS)/sqrt(FeLNS) #-0.183
ResLBR=(FoLBR-FeLBR)/sqrt(FeLBR) #1.094
ResLSG=(FoLSG-FeLSG)/sqrt(FeLSG) #-1.099
```

This did create standardised residuals which I can see. So I am going to plug these into the table I created earlier.

```
std.res.table <- matrix(c(0.136,-0.815,0.819,-0.183,1.094,-1.099), ncol=3, by
row=TRUE)
colnames(std.res.table) <- c('Not Stopped', 'Bribe requested', 'Stopped/Given
warning')
rownames(std.res.table) <- c('Upper Class', 'Lower Class')
std.res.table <- as.table(std.res.table)
std.res.table
```

```
##             Not Stopped Bribe requested Stopped/Given warning
## Upper Class       0.136          -0.815                 0.819
## Lower Class      -0.183           1.094                -1.099
```

The question asked to calculate standardised residuals, so I did not see the need to continue and calculate adjusted residuals the formula for which is:

Adjusted residual = (observed – expected) / √[expected x (1 – row total proportion) x (1-column total proportion)]

Question 1 (d) How might the standardized residuals help you interpret the results?

Standardised residuals would be useful to interpreting the results as they tell you which of the cells is more significant to creating your chi square result. It measures how different the cells' expected value is from the observed value. Small standardised residuals tell us that the prediction line is a good fit for the data.

QUESTION 2

Create a data frame from the raw data given in the question

Tried it like this:

economics <- data.frame(x=c("GP","village","reserved","female","irrigation","water"),
y=c(1,2,1,1,0,10 1,1,1,1,5,0 2,2,1,1,2,2 2,1,1,1,4,31 3,2,0,0,0,0 3,1,0,0,0,0 4,2,0,0,4,7
4,1,0,0,0,12 5,2,0,0,0,28 5,1,0,0,0,0 6,2,0,0,0,23 6,1,0,0,4,12 7,2,0,0,0,0 7,1,0,0,0,0
8,2,1,1,4,41 8,1,1,1,5,23 9,2,0,0,0,0 9,1,0,0,1,0 10,2,0,0,9,12 10,1,0,0,52,59 11,2,0,0,20,70
11,1,0,0,5,7 12,2,0,0,0,2 12,1,0,0,0,3 13,2,0,0,32,23 13,1,0,0,0,0 14,2,1,1,0,1 14,1,1,1,0,1
15,2,1,1,2,0 15,1,1,1,12,0 16,2,0,0,0,5 16,1,0,0,0,6 17,2,0,0,1,10 17,1,0,0,0,3 18,2,1,1,1,7
18,1,1,1,0,10 19,2,0,0,0,41 19,1,0,0,0,76 20,2,1,1,0,0 20,1,1,1,21,123 21,2,0,0,1,1 21,1,0,0,0,7
22,2,0,0,0,0 22,1,0,0,0,0 23,2,0,0,0,2 23,1,0,0,0,18 24,2,0,0,7,18 24,1,0,0,0,16 25,2,1,1,21,340
25,1,1,1,15,6 26,2,0,1,5,5 26,1,0,1,5,0 27,2,1,1,40,309 27,1,1,1,0,4 28,2,0,0,0,5 28,1,0,0,0,8
29,2,0,0,0,9 29,1,0,0,3,0 30,2,1,1,7,31 30,1,1,1,0,200 31,2,0,0,0,20 31,1,0,0,0,12 32,2,0,0,0,28
32,1,0,0,5,41 33,2,0,0,0,0 33,1,0,0,1,0 34,2,1,1,0,10 34,1,1,1,0,5 35,2,0,0,0,2 35,1,0,0,15,4
36,2,0,0,3,59 36,1,0,0,0,19 37,2,1,1,0,8 37,1,1,1,0,1 38,2,0,0,0,10 38,1,0,0,0,4 39,2,0,0,0,16
39,1,0,0,0,2 40,2,0,0,0,5 40,1,0,0,0,3 41,2,0,0,0,10 41,1,0,0,0,3 42,2,0,0,0,9 42,1,0,0,0,6
43,2,1,1,0,1 43,1,1,1,7,7 44,2,1,1,2,25 44,1,1,1,0,0 45,2,0,0,0,11 45,1,0,0,0,0 46,2,1,1,1,13
46,1,1,1,0,4 47,2,0,0,0,17 47,1,0,0,0,17 48,2,1,1,0,5 48,1,1,1,0,8 49,2,1,1,10,78 49,1,1,1,0,19
50,2,0,0,0,2 50,1,0,0,7,41 51,2,1,1,6,0 51,1,1,1,0,3 52,2,0,0,0,8 52,1,0,0,0,5 53,2,0,0,0,21
53,1,0,0,0,6 54,2,0,0,15,26 54,1,0,0,2,9 55,2,0,0,10,8 55,1,0,0,0,3 56,2,0,0,0,0 56,1,0,0,0,1
57,2,1,1,0,9 57,1,1,1,0,10 58,2,0,0,0,11 58,1,0,0,0,12 59,2,1,1,0,0 59,1,1,1,0,5 60,2,0,0,0,24
60,1,0,0,0,24 61,2,0,0,0,0 61,1,0,0,0,10 62,2,1,1,0,0 62,1,1,1,0,144 63,2,1,1,0,2 63,1,1,1,0,5
64,2,0,0,0,6 64,1,0,0,0,1 65,2,1,1,0,43 65,1,1,1,0,9 66,2,1,1,0,38 66,1,1,1,5,25 67,2,0,0,0,11
67,1,0,0,0,15 68,2,0,0,0,0 68,1,0,0,0,9 69,2,0,0,0,30 69,1,0,0,0,2 70,2,1,1,0,98 70,1,1,1,0,44
71,2,0,0,0,1 71,1,0,0,0,5 72,2,1,1,0,2 72,1,1,1,0,4 73,2,0,0,0,10 73,1,0,0,0,14 74,2,0,0,0,0
74,1,0,0,0,0 75,2,0,0,0,3 75,1,0,0,13,23 76,2,0,0,0,1 76,1,0,0,0,10 77,2,0,0,0,5 77,1,0,0,0,9
78,2,1,1,0,6 78,1,1,1,0,2 79,2,0,0,0,0 79,1,0,0,5,6 80,2,1,1,0,8 80,1,1,1,0,7 81,2,0,0,0,0
81,1,0,0,3,4 82,2,0,0,3,11 82,1,0,0,0,17 83,2,1,1,0,0 83,1,1,1,6,13 84,2,0,1,1,9 84,1,0,1,0,5
85,2,0,0,0,0 85,1,0,0,0,4 86,2,0,1,0,12 86,1,0,1,1,13 87,2,0,0,0,4 87,1,0,0,0,28 88,2,0,0,1,1
88,1,0,0,1,13 89,2,0,0,0,8 89,1,0,0,0,0 90,2,1,1,0,3 90,1,1,1,2,6 91,2,1,1,0,3 91,1,1,1,0,16
92,2,0,1,0,54 92,1,0,1,15,2 93,2,0,0,0,8 93,1,0,0,0,23 94,2,0,0,1,9 94,1,0,0,0,12 95,2,1,1,0,11
95,1,1,1,1,5 96,2,0,0,0,16 96,1,0,0,0,2 97,2,1,1,0,9 97,1,1,1,12,15 98,2,1,1,0,12 98,1,1,1,2,8
99,2,0,0,6,13 99,1,0,0,1,7 100,2,1,1,5,2 100,1,1,1,0,7 101,2,0,0,2,20 101,1,0,0,0,3
102,2,1,1,1,60 102,1,1,1,0,6 103,2,0,0,0,3 103,1,0,0,0,3 104,2,0,0,0,22 104,1,0,0,9,7
105,2,0,0,0,13 105,1,0,0,0,22 106,2,0,0,1,8 106,1,0,0,0,14 107,2,0,1,1,2 107,1,0,1,0,24
108,2,0,0,0,13 108,1,0,0,0,21 109,2,1,1,0,5 109,1,1,1,0,8 110,2,0,0,0,24 110,1,0,0,0,7
111,2,1,1,0,18 111,1,1,1,0,29 112,2,0,0,0,26 112,1,0,0,0,10 113,2,0,0,16,40 113,1,0,0,8,10
114,2,1,1,14,7 114,1,1,1,2,16 115,2,1,1,0,7 115,1,1,1,0,5 116,2,0,0,10,12 116,1,0,0,0,4
117,2,0,0,0,20 117,1,0,0,0,9 118,2,1,1,2,71 118,1,1,1,0,5 119,2,0,1,18,4 119,1,0,1,2,17
120,2,0,0,47,47 120,1,0,0,15,9 121,2,0,0,0,25 121,1,0,0,0,62 122,2,0,0,8,58 122,1,0,0,0,6
123,2,1,1,0,5 123,1,1,1,0,15 124,2,0,0,4,8 124,1,0,0,0,12 125,2,1,1,0,32 125,1,1,1,0,17
126,2,0,0,2,10 126,1,0,0,8,4 127,2,0,0,0,3 127,1,0,0,0,3 128,2,1,1,2,16 128,1,1,1,15,3
129,2,1,1,5,86 129,1,1,1,24,11 130,2,1,1,0,21 130,1,1,1,0,5 131,2,0,0,90,54 131,1,0,0,34,8

132,2,0,0,0,19 132,1,0,0,0,4 133,2,0,0,8,77 133,1,0,0,0,44 134,2,1,1,1,4 134,1,1,1,6,4
135,2,1,1,0,26 135,1,1,1,0,10 136,2,0,0,0,21 136,1,0,0,20,46 137,2,0,0,6,30 137,1,0,0,0,20
138,2,0,0,3,10 138,1,0,0,2,2 139,2,0,0,5,90 139,1,0,0,83,155 140,2,0,0,1,23 140,1,0,0,0,8
141,2,1,1,9,20 141,1,1,1,4,22 142,2,0,1,2,25 142,1,0,1,0,20 143,2,1,1,3,21 143,1,1,1,32,59
144,2,0,0,3,8 144,1,0,0,0,1 145,2,0,1,0,25 145,1,0,1,0,4 146,2,0,0,0,7 146,1,0,0,0,40
147,2,1,1,2,7 147,1,1,1,0,0 148,2,0,0,0,19 148,1,0,0,2,3 149,2,0,0,0,40 149,1,0,0,0,2
150,2,1,1,1,17 150,1,1,1,0,9 151,2,0,0,4,7 151,1,0,0,1,42 152,2,0,0,0,1 152,1,0,0,1,3
153,2,0,0,0,3 153,1,0,0,0,66 154,2,0,0,0,6 154,1,0,0,30,35 155,2,0,0,3,51 155,1,0,0,10,13
156,2,0,0,0,7 156,1,0,0,0,16 157,2,0,0,0,20 157,1,0,0,0,21 158,2,0,0,0,3 158,1,0,0,8,4
159,2,1,1,0,11 159,1,1,1,0,29 160,2,1,1,0,2 160,1,1,1,5,5 161,2,0,0,0,2 161,1,0,0,0,11)

```
#This did not work.


#Tried again using read.csv function which worked.


```r
economics<-read.csv("https://raw.githubusercontent.com/kosukeimai/qss/master/
PREDICTION/women.csv")
str(economics)

## 'data.frame':    322 obs. of  6 variables:
##  $ GP        : int  1 1 2 2 3 3 4 4 5 5 ...
##  $ village   : int  2 1 2 1 2 1 2 1 2 1 ...
##  $ reserved  : int  1 1 1 1 0 0 0 0 0 0 ...
##  $ female    : int  1 1 1 1 0 0 0 0 0 0 ...
##  $ irrigation: int  0 5 2 4 0 0 4 0 0 0 ...
##  $ water     : int  10 0 2 31 0 0 7 12 28 0 ...
```

Q2(a) State a null and alternative (two-tailed) hypothesis.

Ho= Villages with reservations for women will have a lower number of new or repaired drinking water facilities.

Ha = Villages with reservations for women will have a higher number of new or repaired drinking water.

Q2(b) Run a bivariate regression to test this hypothesis in R (include your code!).

Regression line using the instruction code given in tutorial four. Dependent variable: water Independent variable: reserved

Need to change reserved to binary variable as data is recognising it as an integer

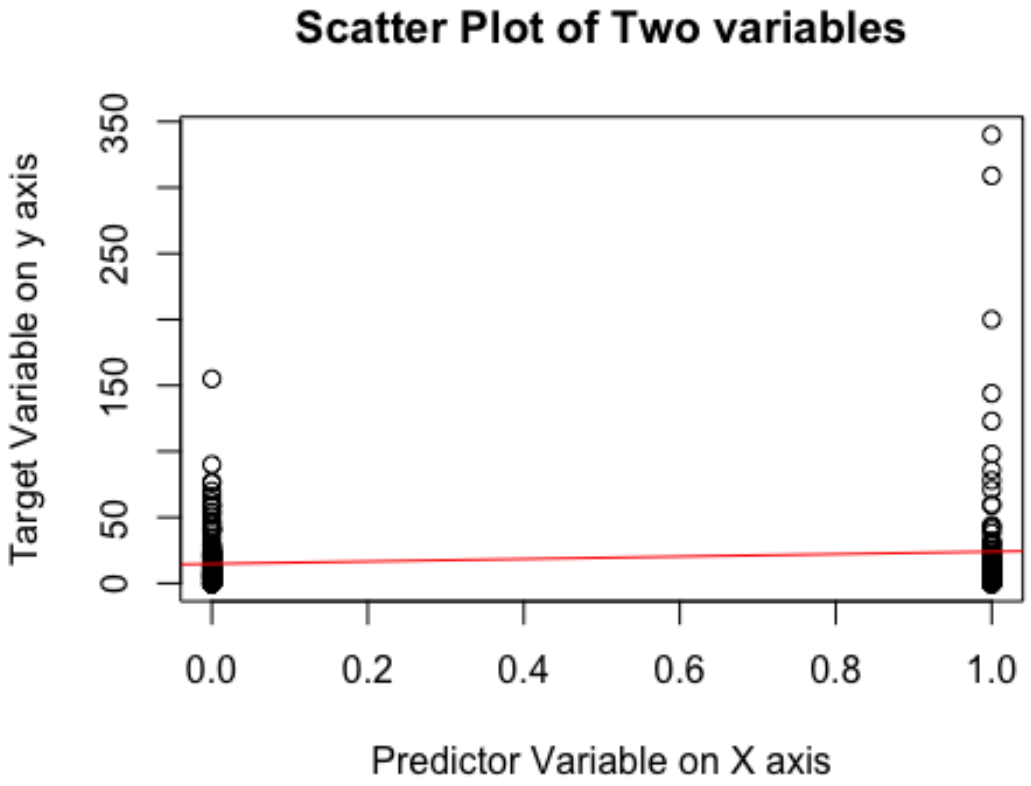
```
plot(economics$reserved, economics$water,
    main = "Scatter Plot of Two variables",
```
```

```
    xlab = "Predictor Variable on X axis",
    ylab = "Target Variable on y axis")
Reserved=as.factor(economics$reserved)

lm(economics$water ~ Reserved)

##
## Call:
## lm(formula = economics$water ~ Reserved)
##
## Coefficients:
## (Intercept)    Reserved1
##      14.738        9.252

#tilde because water measure is dependent on reservation for female politicia
ns
abline(lm(economics$water ~ Reserved), col ="red")
```



Q2(c) Interpret the coefficient estimate for reservation policy.

14.738 (Intercept) is the value of the explained variable (water) while 9.252 is how much the measure of the water variable changes with every one unit increase in the explanatory

variable (reserved). Therefore, for every village with reservations for women, the number of drinking water facilities in villages will increase by 9.252. This means that we can reject the null hypothesis (Ho) and accept the alternative hypothesis (Ha).

QUESTION 3

Q3 (a) Import the data set and obtain summary statistiscs and examine the distribution of the overall lifespan of the fruitflies.

Load dataset, use the str and summary functions to see the information inside the data set. I got the website address from the ASDS group chat students have formed, somebody found it online, because data(fruitfly) would not work for any of us.

```
fruitfly<- read.csv("https://www.zoology.ubc.ca/~bio501/R/data/fruitflies.csv
")
str(fruitfly)
```

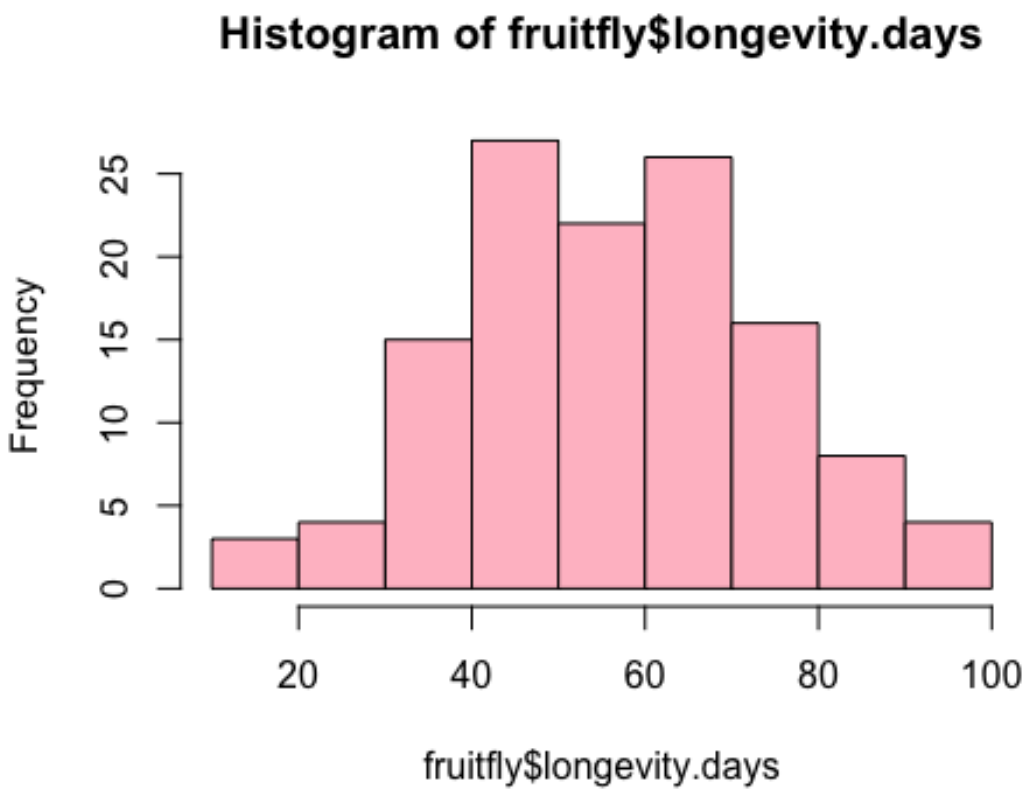
```
## 'data.frame':    125 obs. of  4 variables:
##  $ Npartners     : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ treatment     : chr  "8 pregnant females" "8 pregnant females" "8 pregn
ant females" "8 pregnant females" ...
##  $ longevity.days: int  35 37 49 46 63 39 46 56 63 65 ...
##  $ thorax.mm     : num  0.64 0.68 0.68 0.72 0.72 0.76 0.76 0.76 0.76 0.76
...
```

```
summary(fruitfly)
```

```
##    Npartners    treatment         longevity.days    thorax.mm
##  Min.   :0.0   Length:125         Min.   :16.00   Min.   :0.640
##  1st Qu.:1.0   Class :character   1st Qu.:46.00   1st Qu.:0.760
##  Median :1.0   Mode  :character   Median :58.00   Median :0.840
##  Mean   :3.6                      Mean   :57.44   Mean   :0.821
##  3rd Qu.:8.0                      3rd Qu.:70.00   3rd Qu.:0.880
##  Max.   :8.0                      Max.   :97.00   Max.   :0.940
```

Fruit flies in this dataset lived at least 16 days and at most 97 days. The median lifespan was a 58 days which is very close to the mean lifespan of 57.44 days which may indicate a normal distribution. Which is reinforced by the below histogram.

```
hist(fruitfly$longevity.days, col="pink")
```

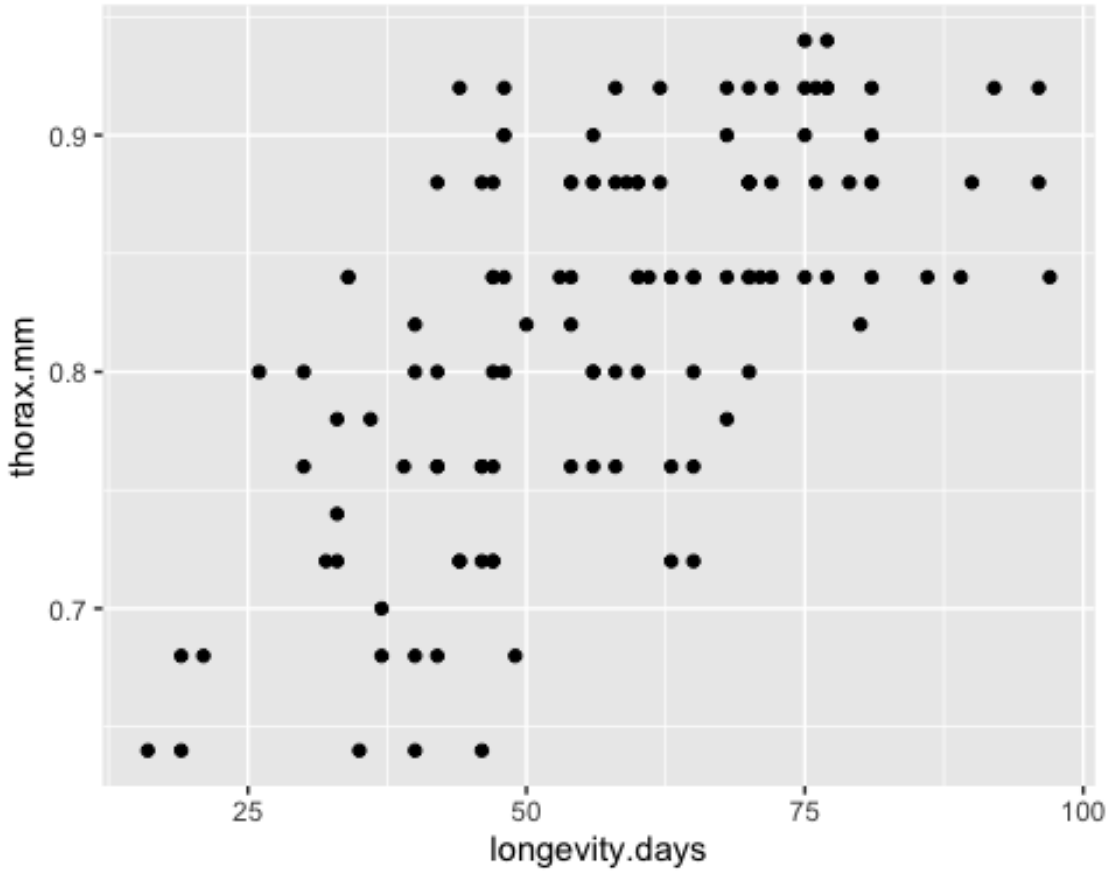


Q3(b) Plot lifespan vs thorax. Does it look like there is a linear relationship? Provide the plot. What is the correlation coefficient between these two variables?

Plot lifespan vs thorax

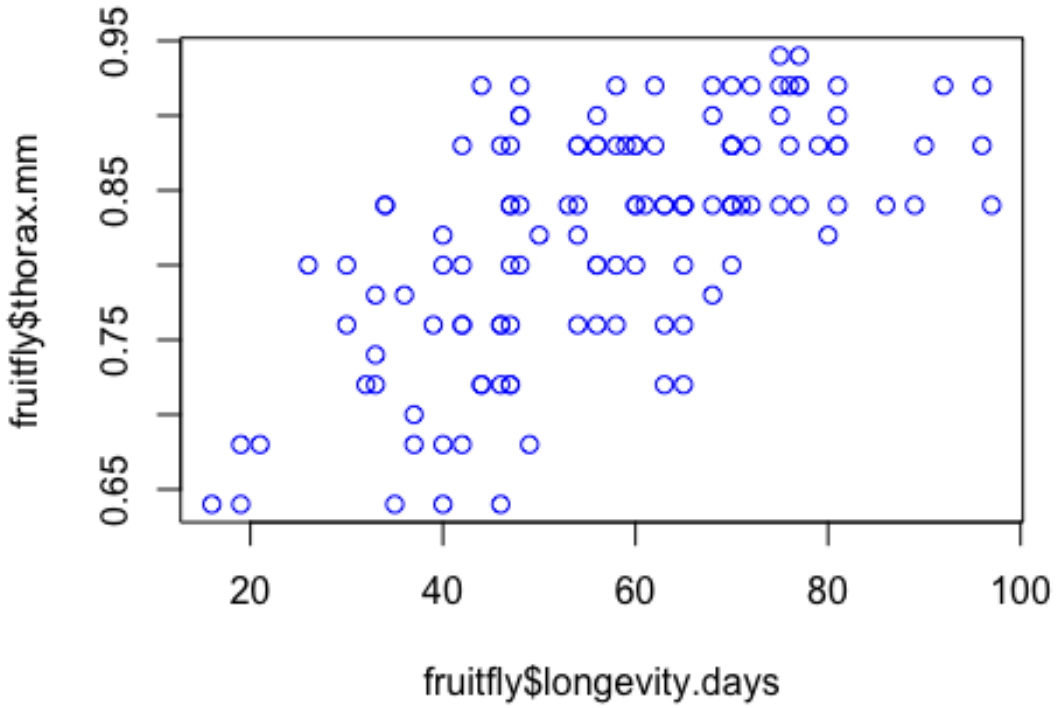
Made a ggplot

```
ggplot(aes(longevity.days, thorax.mm), data=fruitfly) +
  geom_point()
```


thorax.mm
0.9
0.8
0.7
25
50
75
100
longevity.days

Faster plot:

```
plot(fruitfly$longevity.days, fruitfly$thorax.mm, col="blue")
```



There does appear to be a positive linear relationship between lifespan and thorax length.

Trying to find correlation coefficients but do not know how to do that with ggplot so going back to basic plotting code for R and using that.

```
cor(fruitfly$longevity.days, fruitfly$thorax.mm) #0.6364835
```

```
## [1] 0.6364835
```

Correlation co-efficient = 0.6364835

0.64 is enough for me to consider the relationship between lifespan and thorax length to be significant. The thorax distribution looks a bit skewed to the left.

Q3(c) Regress lifespan on thorax. Interpret the slope of the fitted model.

Create a regression

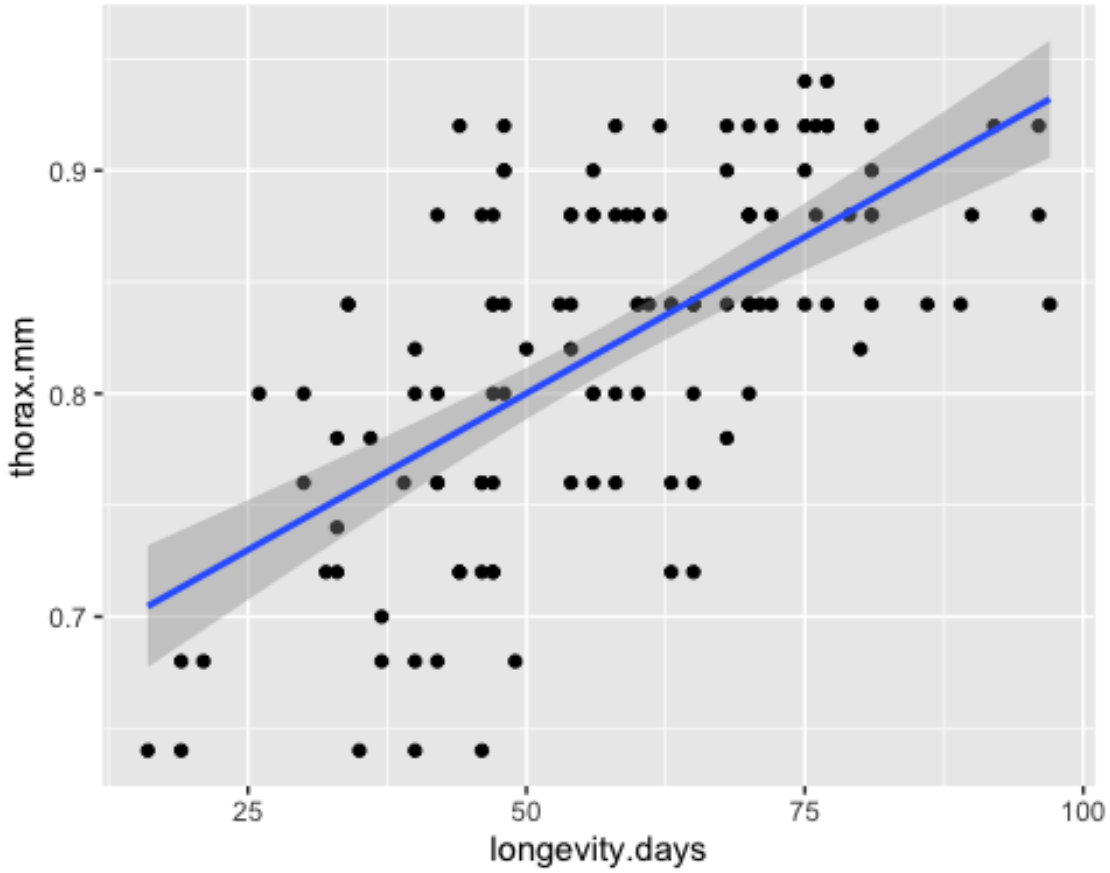
```
regress <- lm(longevity.days ~ thorax.mm, data = fruitfly)
summary(regress)
```

```
##
## Call:
## lm(formula = longevity.days ~ thorax.mm, data = fruitfly)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.415  -9.961   1.132   9.265  36.812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -61.05      13.00  -4.695  7.0e-06 ***
## thorax.mm     144.33      15.77   9.152  1.5e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.6 on 123 degrees of freedom
## Multiple R-squared:  0.4051, Adjusted R-squared:  0.4003
## F-statistic: 83.76 on 1 and 123 DF,  p-value: 1.497e-15
```

For every unit that longevity increases there is an increase in thorax length by 144mm according to the coefficients of this regression. To support my understanding of the regression I will visualise it in a graph.

Visualise in a graph

```
ggplot(aes(longevity.days, thorax.mm), data=fruitfly) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x)
```



There is a an observable positive relationship between lifespan and thorax.

Q3(d) Test for a significant linear relationship between lifespan and thorax. Provide and interpret your results of your test.

Test for significant linear relationship

To test this relationship I will need to extract p-values from the regression to determine significance.

First I need to create a null hypothesis. This hypothesis will be that the slope (beta) of the regression is 0. Ho= beta=0 Ha= beta not = 0

I willl test this by applying the lm() function to the regression and extracting the p-value from the call.

```
regress.lm <- lm(longevity.days ~ thorax.mm , data = fruitfly)
summary(regress.lm)

##
## Call:
## lm(formula = longevity.days ~ thorax.mm, data = fruitfly)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.415  -9.961   1.132   9.265  36.812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -61.05      13.00  -4.695  7.0e-06 ***
## thorax.mm     144.33      15.77   9.152  1.5e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.6 on 123 degrees of freedom
## Multiple R-squared:  0.4051, Adjusted R-squared:  0.4003
## F-statistic: 83.76 on 1 and 123 DF,  p-value: 1.497e-15
```

The p-value from this is: p-value: 1.497e-15

I did have to google what this was scientific notation for and as a real number 1.497e-15 = 0.000000000000001497

As the p-value is significantly lower than the common value of alpha=0.05, we can reject the null hypothesis.

Therefore there value of beta in this regression is highly unlikely to be 0, indicating that there is a significant relationship between thorax length and lifespan in fruit flies.

Q3(e) Provide the 90% confidence interval for the slope of the fitted model.

• Use the formula of confidence interval. Tried looking up formulas for coefficients but all require the mean and when I try to find the mean in R all I get is “NA”

```
summary(regress.lm)

##
## Call:
## lm(formula = longevity.days ~ thorax.mm, data = fruitfly)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.415  -9.961   1.132   9.265  36.812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -61.05      13.00  -4.695  7.0e-06 ***
## thorax.mm     144.33      15.77   9.152  1.5e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 13.6 on 123 degrees of freedom
## Multiple R-squared:  0.4051, Adjusted R-squared:  0.4003
## F-statistic: 83.76 on 1 and 123 DF,  p-value: 1.497e-15
```

```
length(regress.lm)
```

```
## [1] 12
```

```
n = 12
mean(regress.lm)
```

```
## Warning in mean.default(regress.lm): argument is not numeric or logical:
## returning NA
```

```
## [1] NA
```

One of the classmates gave the formula as conf=slope + margin of error margin of error = critical value x standard error

Slope = 0.0028068 (coefficient estimate for x) Margin of error =

Critical value for 90% Alpha level = 100%-90% = 10% = 0.1 Alpha level divided by 2 = two-tailed = 0.05 0.05 from 1 = 0.95 Z table -> 1.645

Standard error thorax: 15.77 Standard error intercept (when thorax=0): 13.00

```
#Confidence Level for thorax:
thoraxconf.upper = 144.33+ (1.645*15.77)
thoraxconf.lower= 144.33 - (1.645*15.77)
cat("90% confidence interval for thorax is", thoraxconf.lower, ",", thoraxcon
f.upper, "\n")
```

```
## 90% confidence interval for thorax is 118.3884 , 170.2717
```

You can find the confidence intervals in the output for the cat code above. Need to find conf for the slope which is beta. Longevity is the beta in the regression.

Answer: 90% confidence interval for thorax is (118.3884 , 170.2717) Note: there could be error due to rounding

• Use the function confint() in R .

```
confint(regress.lm)
```

```
##                 2.5 %   97.5 %
## (Intercept) -86.79221 -35.3112
## thorax.mm   113.11646 175.5497
```

Thorax: 113.11646 , 175.5497

Q3(f) Use the predict() function in R to (1) predict an individual fruitfly's lifespan when thorax=0.8 and (2) the average lifespan of fruitflies when thorax=0.8 by the fitted model. This requires that you compute prediction and confidence intervals.

What are the expected values of lifespan? What are the prediction and confidence intervals around the expected values?

What is the difference between predicted and fitted models?

?predict()

1) predict

```
new.data=data.frame(thorax.mm=0.8)
predict.1=predict(regress.lm, newdata=new.data)
predict.1
```

```
##        1
## 54.41478
```

predict.1= 54.4 When thorax length is 0.8mm, the fruit fly is predicted to live to 54.4 days old.

2) fitted model

```
fitted = fitted(regress.lm)
dat = cbind(fruitfly$thorax.mm, unname(fitted))
dat
```

```
##        [,1]     [,2]
##   [1,] 0.64 31.32148
##   [2,] 0.68 37.09481
##   [3,] 0.68 37.09481
##   [4,] 0.72 42.86813
##   [5,] 0.72 42.86813
##   [6,] 0.76 48.64145
##   [7,] 0.76 48.64145
##   [8,] 0.76 48.64145
##   [9,] 0.76 48.64145
##  [10,] 0.76 48.64145
##  [11,] 0.80 54.41478
##  [12,] 0.80 54.41478
##  [13,] 0.80 54.41478
##  [14,] 0.84 60.18810
##  [15,] 0.84 60.18810
##  [16,] 0.84 60.18810
##  [17,] 0.84 60.18810
##  [18,] 0.84 60.18810
##  [19,] 0.84 60.18810
##  [20,] 0.88 65.96143
##  [21,] 0.88 65.96143
##  [22,] 0.92 71.73475
##  [23,] 0.92 71.73475
##  [24,] 0.92 71.73475
##  [25,] 0.94 74.62141
##  [26,] 0.64 31.32148
##  [27,] 0.70 39.98147
```

```
##  [28,] 0.72 42.86813
##  [29,] 0.72 42.86813
##  [30,] 0.72 42.86813
##  [31,] 0.76 48.64145
##  [32,] 0.78 51.52812
##  [33,] 0.80 54.41478
##  [34,] 0.84 60.18810
##  [35,] 0.84 60.18810
##  [36,] 0.84 60.18810
##  [37,] 0.84 60.18810
##  [38,] 0.84 60.18810
##  [39,] 0.88 65.96143
##  [40,] 0.88 65.96143
##  [41,] 0.88 65.96143
##  [42,] 0.88 65.96143
##  [43,] 0.88 65.96143
##  [44,] 0.92 71.73475
##  [45,] 0.92 71.73475
##  [46,] 0.92 71.73475
##  [47,] 0.92 71.73475
##  [48,] 0.92 71.73475
##  [49,] 0.92 71.73475
##  [50,] 0.94 74.62141
##  [51,] 0.64 31.32148
##  [52,] 0.68 37.09481
##  [53,] 0.72 42.86813
##  [54,] 0.76 48.64145
##  [55,] 0.76 48.64145
##  [56,] 0.80 54.41478
##  [57,] 0.80 54.41478
##  [58,] 0.80 54.41478
##  [59,] 0.82 57.30144
##  [60,] 0.82 57.30144
##  [61,] 0.84 60.18810
##  [62,] 0.84 60.18810
##  [63,] 0.84 60.18810
##  [64,] 0.84 60.18810
##  [65,] 0.84 60.18810
##  [66,] 0.84 60.18810
##  [67,] 0.88 65.96143
##  [68,] 0.88 65.96143
##  [69,] 0.88 65.96143
##  [70,] 0.88 65.96143
##  [71,] 0.88 65.96143
##  [72,] 0.88 65.96143
##  [73,] 0.88 65.96143
##  [74,] 0.92 71.73475
##  [75,] 0.92 71.73475
##  [76,] 0.68 37.09481
##  [77,] 0.68 37.09481
```

```
##  [78,] 0.72 42.86813
##  [79,] 0.76 48.64145
##  [80,] 0.78 51.52812
##  [81,] 0.80 54.41478
##  [82,] 0.80 54.41478
##  [83,] 0.80 54.41478
##  [84,] 0.84 60.18810
##  [85,] 0.84 60.18810
##  [86,] 0.84 60.18810
##  [87,] 0.84 60.18810
##  [88,] 0.84 60.18810
##  [89,] 0.84 60.18810
##  [90,] 0.88 65.96143
##  [91,] 0.88 65.96143
##  [92,] 0.88 65.96143
##  [93,] 0.90 68.84809
##  [94,] 0.90 68.84809
##  [95,] 0.90 68.84809
##  [96,] 0.90 68.84809
##  [97,] 0.90 68.84809
##  [98,] 0.90 68.84809
##  [99,] 0.92 71.73475
## [100,] 0.92 71.73475
## [101,] 0.64 31.32148
## [102,] 0.64 31.32148
## [103,] 0.68 37.09481
## [104,] 0.72 42.86813
## [105,] 0.72 42.86813
## [106,] 0.74 45.75479
## [107,] 0.76 48.64145
## [108,] 0.76 48.64145
## [109,] 0.76 48.64145
## [110,] 0.78 51.52812
## [111,] 0.80 54.41478
## [112,] 0.80 54.41478
## [113,] 0.82 57.30144
## [114,] 0.82 57.30144
## [115,] 0.84 60.18810
## [116,] 0.84 60.18810
## [117,] 0.84 60.18810
## [118,] 0.84 60.18810
## [119,] 0.88 65.96143
## [120,] 0.88 65.96143
## [121,] 0.88 65.96143
## [122,] 0.88 65.96143
## [123,] 0.88 65.96143
## [124,] 0.88 65.96143
## [125,] 0.92 71.73475
```

```
colnames(dat) = c("Thorax.mm", "Fitted")
dat = as.data.frame(dat)
dat.08 = dat[which(dat$Thorax == 0.8), ]

mean(dat.08$Fitted)

## [1] 54.41478
```

54.41478 Find prediction and confidence intervals

```
predict(regress.lm, newdata = new.data, interval = 'confidence')

##        fit      lwr      upr
## 1 54.41478 51.91932 56.91024
```

fit lwr upr 1 54.41478 51.91932 56.91024

```
predict(regress.lm, newdata = new.data, interval = 'prediction')

##        fit      lwr      upr
## 1 54.41478 27.37542 81.45414

   fit      lwr      upr
```

1 54.41478 27.37542 81.45414

Prediction interval takes into account the variability and error of the estimators so it is wider.

Average lifespan is the same for both confidence and prediction interval. Prediction interval has larger interval so expected lifespan is between 27 days and 81 days.

Q3(g) For a sequence of thorax values, draw a plot with their fitted values for lifespan, as well as the prediction intervals and confidence intervals.

Create a sequence of the thorax values for the new plot

```
index = seq(from = 0, to = 125, by= 5)
#takes every fifth number
s= dat[index,]

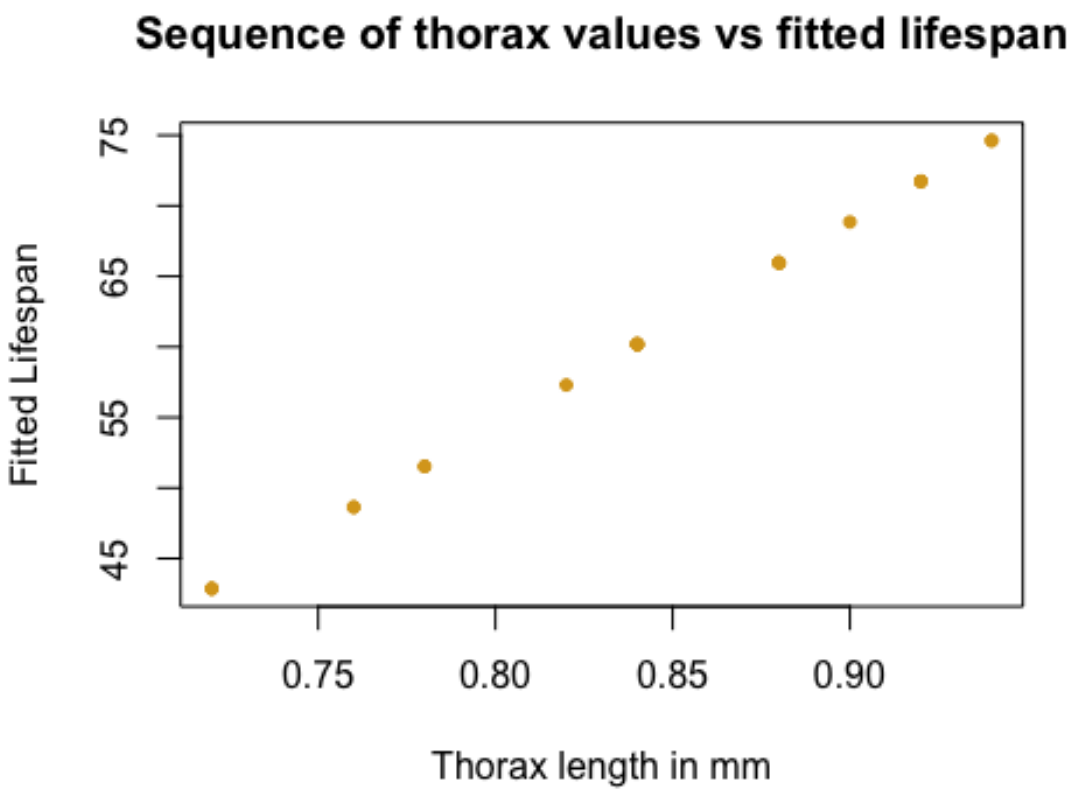
ndata = s[,1]
d = unname(ndata)
conf = predict(regress.lm, newdata =data.frame(thorax.mm=d), interval="confid
ence")
pred = predict(regress.lm, newdata =data.frame(thorax.mm=d), interval="predic
tion")
pred

##         fit      lwr       upr
## 1  42.86813 15.65358  70.08268
```

```
## 2  48.64145 21.54311  75.73980
## 3  60.18810 33.15012  87.22608
## 4  65.96143 38.86723  93.05563
## 5  74.62141 47.33574 101.90708
## 6  42.86813 15.65358  70.08268
## 7  60.18810 33.15012  87.22608
## 8  65.96143 38.86723  93.05563
## 9  71.73475 44.52708  98.94242
## 10 74.62141 47.33574 101.90708
## 11 48.64145 21.54311  75.73980
## 12 57.30144 30.26998  84.33290
## 13 60.18810 33.15012  87.22608
## 14 65.96143 38.86723  93.05563
## 15 71.73475 44.52708  98.94242
## 16 51.52812 24.46645  78.58978
## 17 60.18810 33.15012  87.22608
## 18 65.96143 38.86723  93.05563
## 19 68.84809 41.70427  95.99191
## 20 71.73475 44.52708  98.94242
## 21 42.86813 15.65358  70.08268
## 22 51.52812 24.46645  78.58978
## 23 60.18810 33.15012  87.22608
## 24 65.96143 38.86723  93.05563
## 25 71.73475 44.52708  98.94242
```

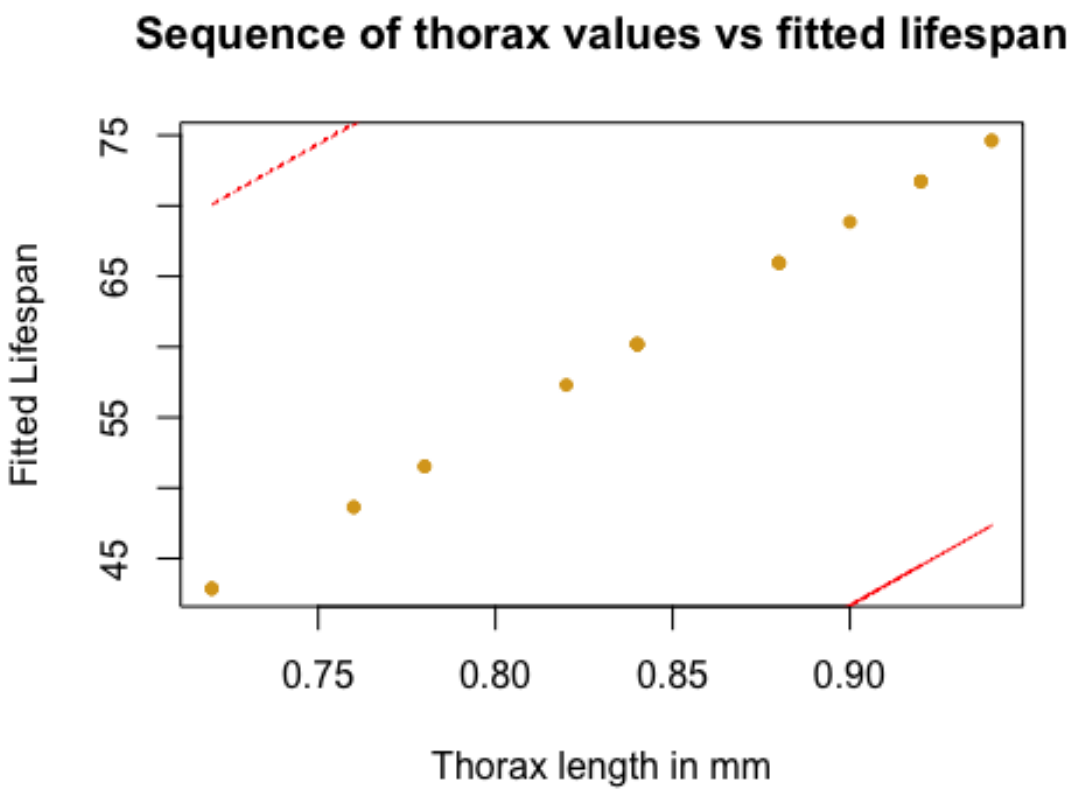
Plot the graph and label it

```
plot(s$Thorax.mm, s$Fitted, pch=20, col="goldenrod",
main="Sequence of thorax values vs fitted lifespan",
xlab= "Thorax length in mm",
ylab= "Fitted Lifespan")
```



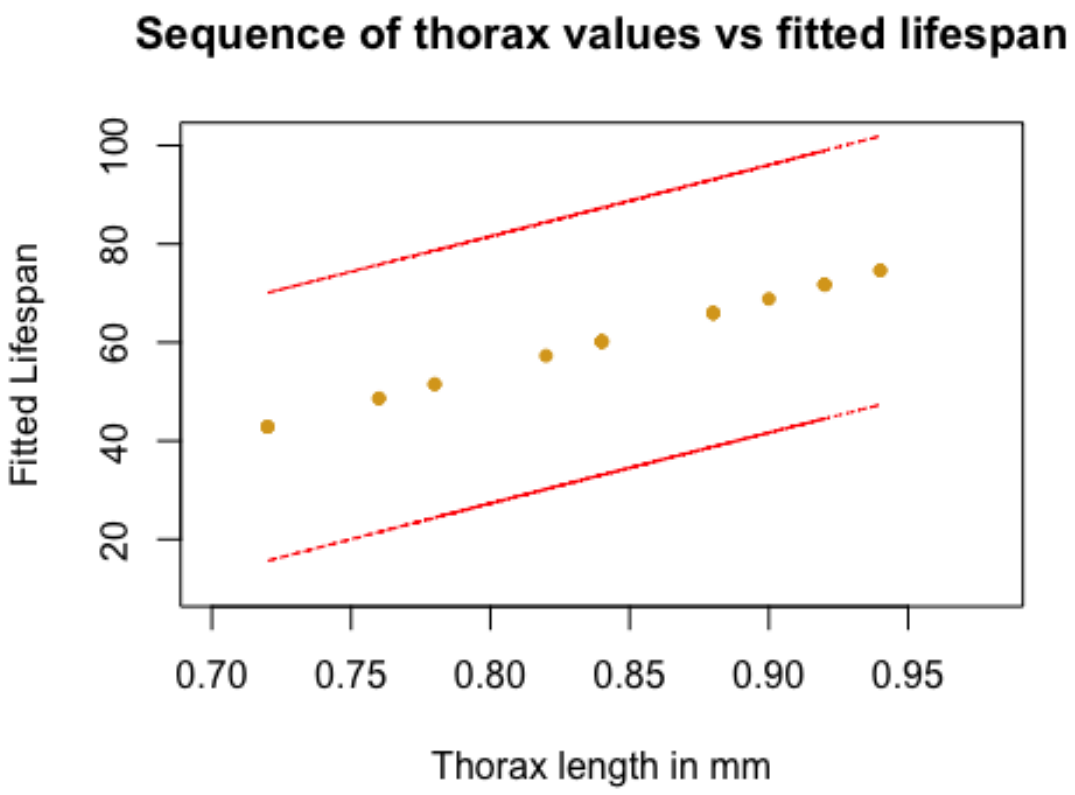
Prediction interval and confidence interval need to add lines

```
plot(s$Thorax.mm, s$Fitted, pch=20, col="goldenrod",
main="Sequence of thorax values vs fitted lifespan",
xlab= "Thorax length in mm",
ylab= "Fitted Lifespan")
lines(s$Thorax.mm, pred[, "lwr"], lty=3, col="red")
lines(s$Thorax.mm, pred[, "upr"], lty=3, col="red")
```



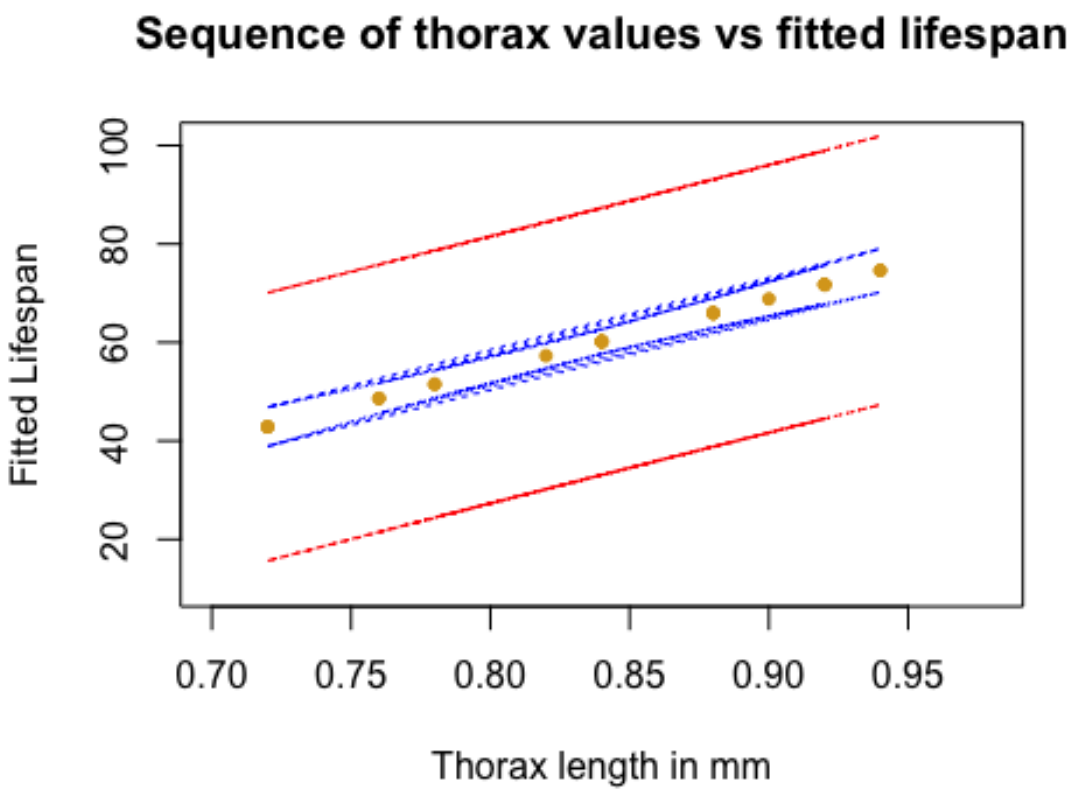
The lines are really weird. I reached out to a friend who understands stats and she explained that the axes were probably in the wrong order. She suggested a fix using xlim and ylim.

```
plot(s$Thorax.mm, s$Fitted, pch=20, col="goldenrod",
main="Sequence of thorax values vs fitted lifespan",
xlab= "Thorax length in mm",
ylab= "Fitted Lifespan",
xlim= range(0.7, 0.98),
ylim= range(10, 101))
lines(s$Thorax.mm, pred[, "lwr"], lty=3, col="red")
lines(s$Thorax.mm, pred[, "upr"], lty=3, col="red")
```



Red lines tracking prediction interval for the fitted values.

```
plot(s$Thorax.mm, s$Fitted, pch=20, col="goldenrod",
main="Sequence of thorax values vs fitted lifespan",
xlab= "Thorax length in mm",
ylab= "Fitted Lifespan",
xlim= range(0.7, 0.98),
ylim= range(10, 101))
lines(s$Thorax.mm, pred[, "lwr"], lty=3, col="red")
lines(s$Thorax.mm, pred[, "upr"], lty=3, col="red")
lines(s$Thorax.mm, conf[, "lwr"], lty=3, col="blue")
lines(s$Thorax.mm, conf[, "upr"], lty=3, col="blue")
```



Blue lines are the confidence intervals.