Statistical Models Homework 1

Sourabh Prakash and Priyanshi Shah

2023-01-20

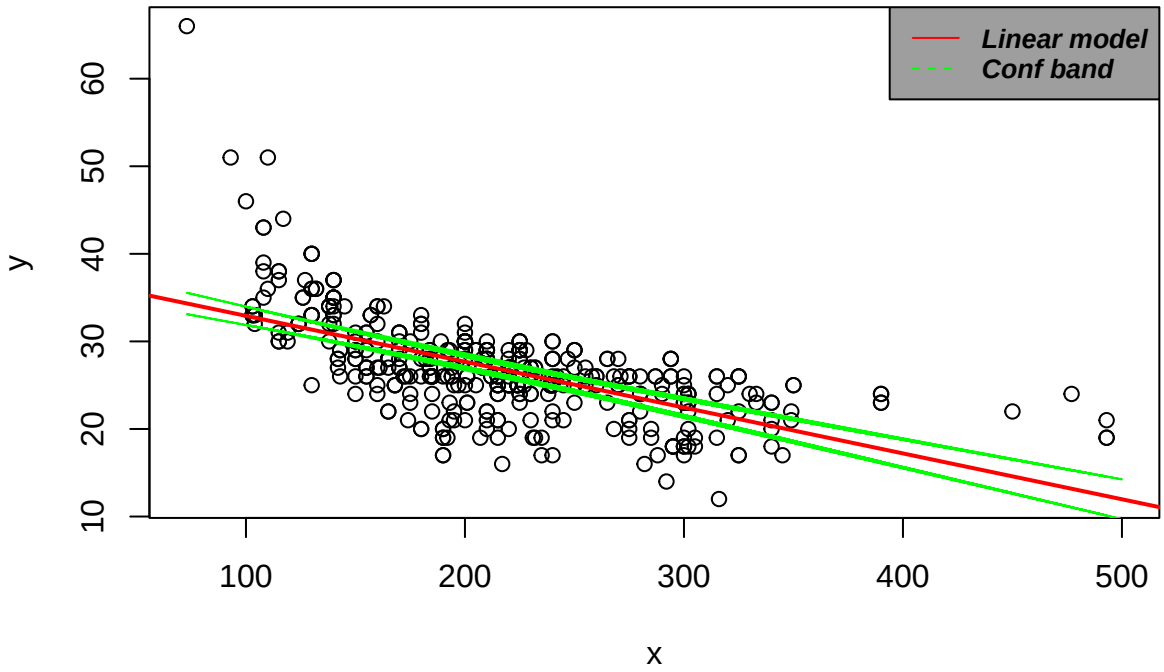
```
#To set path to the working directory, please paste your path below where 04cars.rda
#setwd("D:/projects/R_Projects/Sandbox/Stats_model_assignments")
#getwd()
```

Question 1

Write a function confBand(x, y, conf=0.95) taking in a predictor vector (x1, . . . , xn) and a response vector y = (y1,. . . ,yn) and return a plot with the points (x1,y1),. . . ,(xn,yn), the least squares line, and the confidence band at level conf. Apply your function to hp and mpg from the 04cars dataset.

```
#Loading the cars dataset
load("04cars.rda")
data = dat[,c(13,15)]
names(data) = c("hp", "mpg")

#defining the confBand function
confBand <- function(x,y, conf = 0.95){

  n = 100
  #Fitting the linear model
  linear_model = lm(y~x)
  p=1
  df_1 = p+1 #degree of freedom 1
  df_2=n-p-1  #degree of freedom 2
  fquartile = sqrt((p+1) * qf(conf, df_1,df_2))

  #Calculating y bar
  y_ = predict(linear_model, data.frame(x=x), se =  T)

  #Calculating the upper and lower confidence bounds
  cb_upper =  (y_$fit + ((fquartile)*y_$se.fit))
  cb_lower =  (y_$fit - ((fquartile)*y_$se.fit))

  #Plotting the conf band and linear model
  plot(x,y, type = 'p')

  lines(x,cb_upper, type = 'l', col ="green")
  lines(x,cb_lower, type = 'l', col ="green")
  abline(linear_model, col = 'red', lwd = 2)
  legend("topright", legend=c("Linear model", "Conf band"),
       col=c("red", "green"), lty=1:2, cex=0.8, text.font=4, bg='001000')
```

```
  #return (list(upb = cb_upper, lwb = cb_lower, y_pred = y_$fit, pred_se = y_$se.fit, Fvalue = fquartil
}

confBand(data$hp, data$mpg)
```



Question 2

Let n = 100 and draw x1. . . .xn~ Unif(0,1) which stay fixed in what follows. Repeat the following experiment N = 1000 times. Generate yi Compute the 99% confidence band and record whether it contains the true line, or not. Summarize the result of this numerical experiment by returning the proportion of times (out of N) that the confidence band contained the true line.

```
   #Uniform distribution U(0,1)
   x = runif(100, 0,1)
  #y_true
   y_true = 1.000+x
   contains_true_line = rep(NA, 1000)
   #iterating 1000 times
   for (j in 1:1000){
     #error term N(0,0.2)
     e = rnorm(100, 0, 0.2)
     y = 1.000+x+e
     #Confidence Interval for each iteration
     CI = data.frame(lower = rep(NA,100),upper = rep(NA,100),contain_true_value = rep(NA,100))
```

```
    #Fitting the linear model
    linear_model= lm(y~x)

    n = 100
    p=1
    df_1 = p+1 #degree of freedom 1
    df_2=n-p-1 #degree of freedom 2
    conf = 0.99
    F_quartile = sqrt((p+1) * qf(conf, df_1,df_2))
    pred_op = predict(linear_model, data.frame(hp=x), se =  T)


    CI$lower = (pred_op$fit - ((F_quartile)*pred_op$se.fit))
    CI$upper = (pred_op$fit + ((F_quartile)*pred_op$se.fit))
    CI$contain_true_value = (CI$lower <=y_true & CI$upper >= y_true)

    #contains_true_line[j] = ((mean(CI$contain_true_value)) ==1)
    contains_true_line[j] = ((sum(CI$contain_true_value)) ==n)
  }
  answer = (sum(contains_true_line)/1000)
  sprintf("Result of the experiment is %f", answer)
```

```
## [1] "Result of the experiment is 0.993000"
```

Contributions:

Both the team members Sourabh Prakash and Priyanshi Shah have contributed equally to the homework by discussing the key points and logic together and doing pair programming. For the implementation part question 1 was contributed by Sourabh Prakash and question 2 by Priyanshi Shah.