Assignment IV

Exercise 1:

A)

Code:

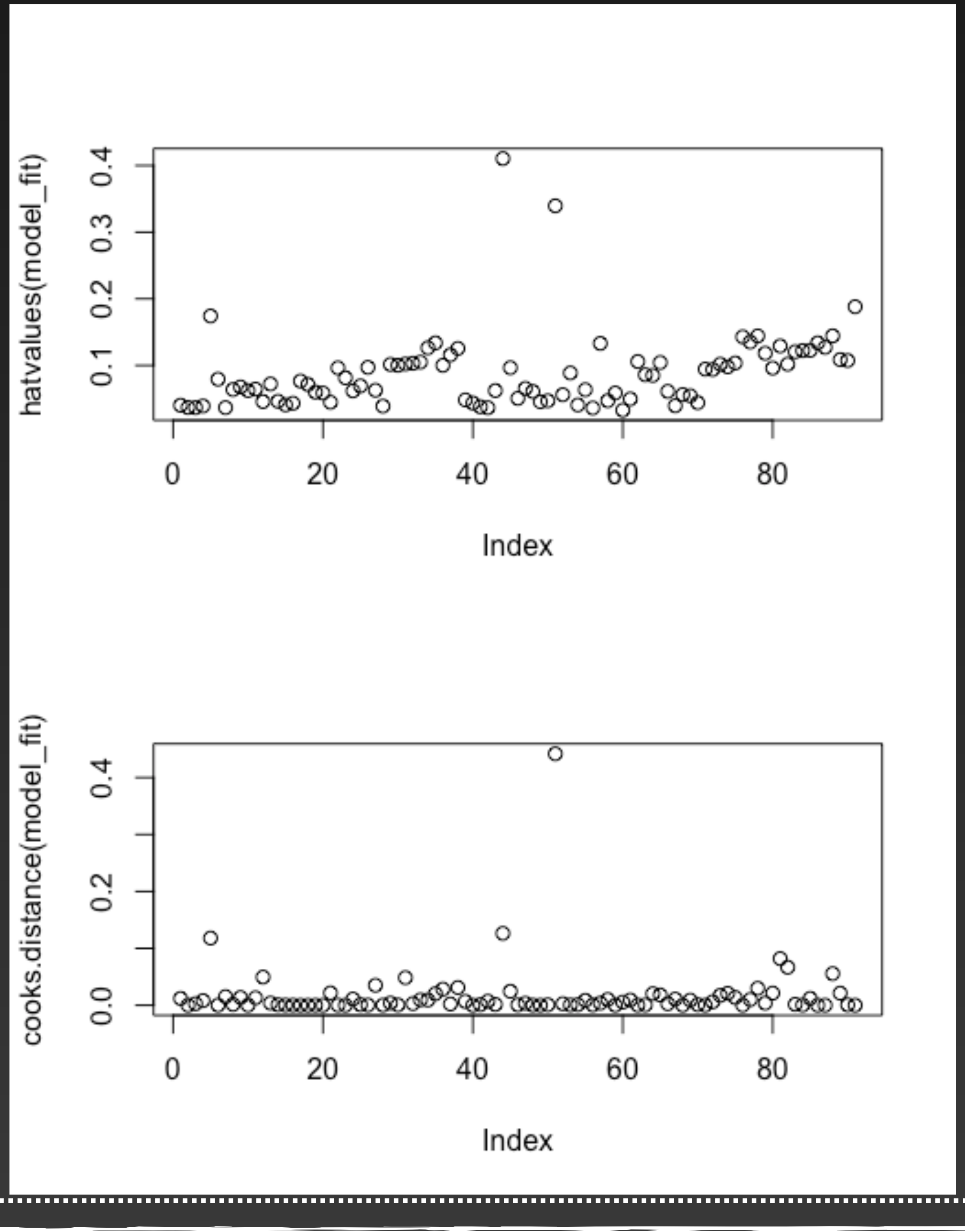
```
darts <- read.csv("Darts.csv")
attach(darts)
name <- factor(Name)
model_ini <- lm(Width~Length+Thickness+name)
model1<-lm(Width~Length+Thickness+name+Length*Thickness)
anova(model_ini,model1)
summary(model1)#Adjusted Rsquared:  0.7062
library(MASS)
b=boxcox(model_ini)#Construct the boxcox to find the fittest λ
I = which(b$y==max(b$y))#Get the highest point
Lamda <- b$x[I]#Get the X value in highest point
model_fit <- lm(Width^Lamda~Length+Thickness+name)
summary(model_fit)
step(model_fit)
par(mfrow=c(2,2))
plot(model_fit)#There are not obvious outlier in it
plot(hatvalues(model_fit))
plot(cooks.distance(model_fit))
```

Output:

```
anova(model_ini,model1)
Analysis of Variance Table
Model 1: Width ~ Length + Thickness + name
Model 2: Width ~ Length + Thickness + name + Length * Thickness
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     84 704.15
2     83 648.28  1    55.865 7.1525 0.009014 **
summary(model_fit)
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.565e-01  1.480e-02  10.575  < 2e-16 ***
Length           -1.386e-03  3.063e-04  -4.524 2.00e-05 ***
Thickness        -8.354e-03  2.071e-03  -4.033 0.000122 ***
nameEnsor        -1.003e-02  2.349e-03  -4.271 5.15e-05 ***
namePedernales   -1.134e-02  2.329e-03  -4.870 5.27e-06 ***
nameTravis        9.406e-05  2.698e-03   0.035 0.972279
nameWells        -7.354e-03  2.642e-03  -2.783 0.006661 **
Length:Thickness  1.325e-04  3.903e-05   3.395 0.001055 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.006255 on 83 degrees of freedom
Multiple R-squared:  0.7723,          Adjusted R-squared:  0.7531
F-statistic: 40.21 on 7 and 83 DF,  p-value: < 2.2e-16

step(model_fit)
Start:  AIC=-915.92
Width^Lamda ~ Length + Thickness + name + Length * Thickness

                   Df  Sum of Sq       RSS      AIC
<none>                            0.0032471 -915.92
- Length:Thickness  1 0.00045089 0.0036980 -906.08
- name              4 0.00190304 0.0051502 -881.94
plot(hatvalues(model_fit)
plot(cooks.distance(model_fit))
```



In this set of data, it contains a categorical variable "name", so first we need to use the factor function to force it to be factor. Next, we can create a simple linear model which only have one item, whose name is "model_ini".

Then I will also create a model named model1 which contain the iteration item. Secondly, I will use the anova function to check the P value. By the output, we can see the P is value is 0.009014, which is smaller than 0.05. That means we will reject the H0 to accept the H1. By the summary function, we can check less variable have three stars and the R^2 only has 0.7062. So we have to optimize the model to make it more consistent with the initial data. Then I hope to use the boxcox to let the model better. By the calculation of boxcox to find the fittest value Lamda. After that, we will let Lamda become the index of response variable and create a new model named "model_fit". Through the summary function, we can see that almost all variables have stars, which means significant, and the adjustment R-square is also increased to 0.7531 from 0.7062, which is improved compared with the previous model. The step function shows the AIC is -915.92 which is too small. Finally, there are not significance outlier in the diagnostic plot. In the end we will choose the model_fit "lm(formula = Width^Lamda ~ Length + Thickness + name + Length * Thickness)" is final choice. However, form the graph of hatvalue and cook distance, there are some outlier points because they have a very high value, which will effect the predict fit value.

B)

Code:

```
data_value <- data.frame(Length=50,Thickness=8,name="Travis")
value <- predict(model_fit,data_value)
predict_value<- value^(1/Lamda)
predict_value#20.18561
```

Firstly, store all known explanatory variables in the data.frame, then, we use the predict function to get the predicted value of the model. But we still need to change that by "value^(1/Lamda)" . In the end we can know when length=50,thickness=8 and name is Travis, the predict value of Width of darts is 20.18561 in our model.

KING'S College LONDON

Assignment IV

Exercise 2:

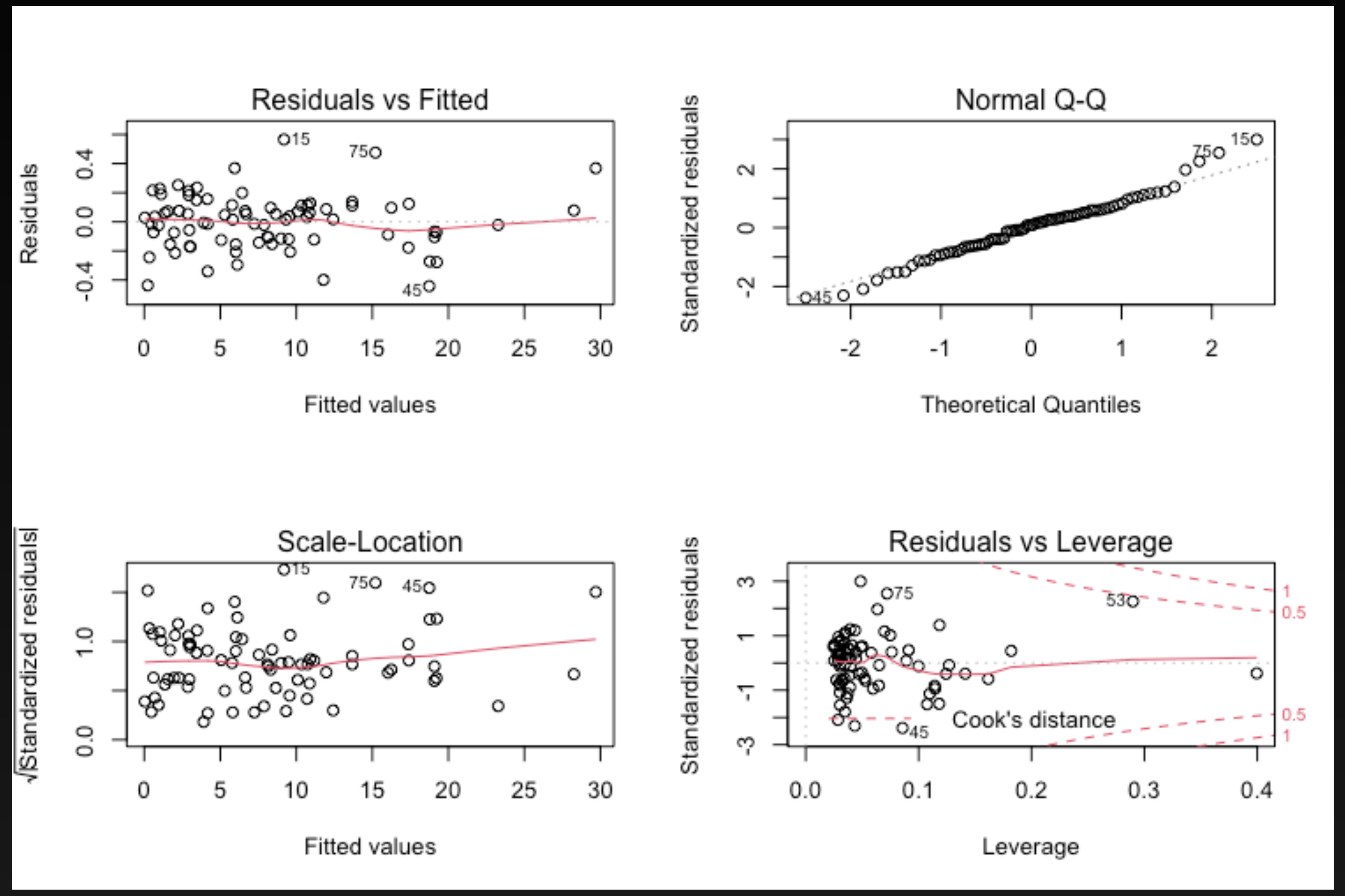
A)

Code:

```
optimal <- read.csv("optimal.csv")
attach(optimal)
model1 <- lm(y~x1+x2+I(x1^2)+I(x2^2))
#Create a model which include all the thing
summary(model1)#Multiple R-squared:  0.9992,Adjusted R-squared:  0.9992
step(model1)# AIC=-257.87
model2 <- lm(y~x1+x2)#A model only have one time item
model3 <- lm(y~I(x1^2)+I(x2^2))#Only have two time item
#By the anova function to compare model 2, model 3 with model 1
anova(model1,model2)#P = 2.2e-16
anova(model1,model3)#P = 2.2e-16
model4<- lm(y~x1+x2+I(x1^2)+I(x2^2)+x1*x2)
anova(model1,model4)#P=0.6133
par(mfrow=c(2,2))
plot(model1)
#Now we need to compute the Mallow CP to check our model again
s<-summary(model1)$sigma
n<-length(y)
df<-summary(model1)$df[2]
Cp<-(df)*(summary(model1)$sigma^2)/s^2+2*(n-df)-n
Cp#5
beta_parameter <- model1$coefficients
b_small<- beta_parameter[2:3]
B_big<-matrix(0,2,2)
B_big[1,1]<- beta_parameter[4]
B_big[2,2]<- beta_parameter[5]
x_opt<–0.5*solve(B_big)%*%b_small
x_opt
eigen(B1)$values#justify the x is minimum or maximum
```

Some important Output about code

The output of plot(model1)



The output of x_opt

```
x1=0.7285732
x2=0.2206449
```

The output of eigen(B1)$values

```
1.9420071
0.1986683
```

In this problem, we need to determine an optimal model, and then use the optimal model to find the corresponding x1, x2 when y reaches the minimum value of the appropriate model. So I will use the RSM to find it. Firstly, I will create a model1 which include all the one time term and two time term. By the summary function and step function, three stars for each parameter are strongly correlated with the model, the Adjusted R-squared is 0.9992 which is close to 1 very much. The AIC is -257.87 is also very small. The diagnostic plots of model1 are fine, there are not significant outlier. Next, we need to check whether all the primary and secondary terms are meaningful to the model. By anova function, the P value are both smaller than 0.05, which mean we do not have the reason to reject the H1 hypothesis. But anova(model1,model4) the P value is 0.6133 which is bigger than 0.05, we must reject H1, which mean the iteration item is not significant. The Mallow CP of model1 is 5 which is not big so we will use model1 as the final estimation model.

To calculate minimum expected y of model1, we can write RSM model we need to create two matrices named B and b. Small b is all the coefficient of one time item. The big B is a matrix of 2*2. Because the model 1 does not have the iteration item, so the matrix B is a diagnostic matrix, the B(1,1) is β11 and B(2,2) is β22 and other are both 0. In the end, we can use the stationary point formula x = -1/2(B^-1)b to get the corresponding solutions of x1 and x2.

x1=0.7285732

x2=0.2206449

Next, we still need to check whether the stationary point is maximum or minimum or not. We need to get the eigen value of B.

1.9420071 and 0.1986683

They are both positive so the stationary is minimum. Finally, when x1=0.7285732 and x2=0.2206449, the expected y will get the minimum.

B)

Code:

```
minvalue <- data.frame(x1=0.7241139,x2=0.2224978)
predict(model1,minvalue,interval = "confidence",level = 0.95)
```

Output:

Fit	Lwr	Upr
-0.1546188	-0.256	-0.05323773

In this question, we need to find the 95% confidence interval of response variable y when x1 and x2 take the values in the previous question respectively. In the last question, we have known x1 = 0.7285732 and x2 = 0.2206449. Next, we need to create a data.frame named "minvalue" to save them. Then, using the predict function whose interval is "confidence" and level is 0.95. We will get the 95% confident interval is (-0.256,-0.05323773)

Assignment IV

Exercise 3:

A)

Code:

```
library(datasets)
data("warpbreaks")
glm1 <- glm(breaks ~ wool*tension, family=poisson,data =
warpbreaks)
summary(glm1)
deviance(glm1)/df.residual(glm1)#3.798024
coef(glm1)
```

In this problem, we need to explain the model assumption in the above code, the relationship between the expected value and the expected value of fracture times. And what are the estimates of all the parameters.

Output

```
Call:
glm(formula = breaks ~ wool * tension, family = poisson(link = "log"),
    data = warpbreaks)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3383  -1.4844  -0.1291   1.1725   3.5153

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.79674    0.04994  76.030  < 2e-16 ***
woolB        -0.45663    0.08019  -5.694 1.24e-08 ***
tensionM     -0.61868    0.08440  -7.330 2.30e-13 ***
tensionH     -0.59580    0.08378  -7.112 1.15e-12 ***
woolB:tensionM  0.63818    0.12215   5.224 1.75e-07 ***
woolB:tensionH  0.18836    0.12990   1.450    0.147
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
Null deviance: 297.37  on 53  degrees of freedom
Residual deviance: 182.31  on 48  degrees of freedom
AIC: 468.97

Number of Fisher Scoring iterations: 4
```

In the above code, we use glm function to create a generalized linear model to fit our data. The model is independent, normality and general linear. The variable of number of break is response variable. And the wool and tension are explanatory variable. Because wool and tension are categorical variable, in the output of summary function we also can know the fundamental value are "A" and "L".

In the model, we assume that the number of breaks is significantly related to the interaction item between wool and tension breaks~wool*tension. And in this hypothesis, we use Poisson probability distribution family. The link function of Poisson is "log". So the expected value of the number of break is equal to the exp(predictors). Next, because deviance(glm1)/df.residual(glm1) = 3.798 which is bigger than 1, so this model exists excessive departure.

By the coef() function, we will get every parameter easily. I will show it in the table below.

Intercept	woolB	tensionM	tensionH	woolB:tensionM	woolB:tensionH
3.79674	-0.45663	-0.61868	-0.59580	0.63818	0.18836

B)

Code:

```
datavalue <- data.frame(wool="A",tension="M")
predict(glm1,datavalue)#3.178054
exp(predict(glm1,datavalue))#24
```

In this problem, we need to calculate the expected number of breaks when the wool type is A and the tension level is M.

Firstly, we need to create a data.frame named datavalue to save the value of wool and tension.

Next, we need to use the predict function to use the model glm1 to calculate the predict value of number of break in Poisson family.

Finally, we cannot forget the expected value and predictors is exponential. Thus, we should use the exp() function to transfer the result.

In the end, we will get the number of break is 24 if the kind wool is A and the level of tension is M.

Thank you for your watching!