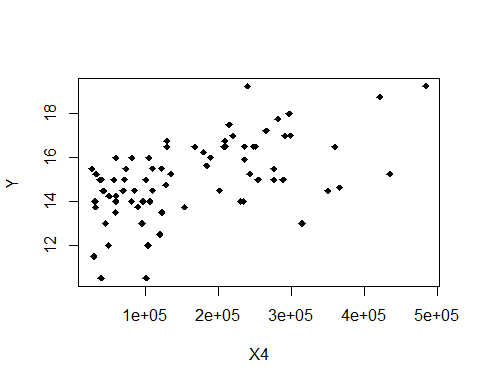
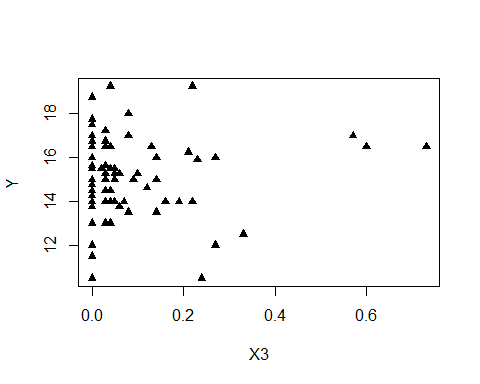
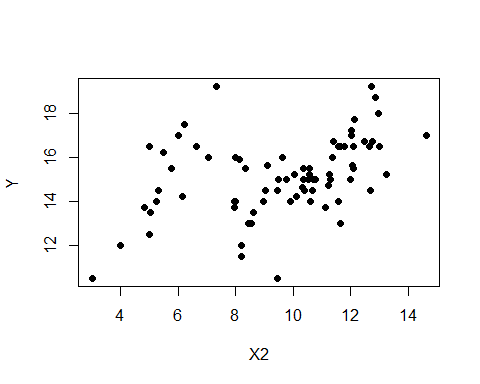
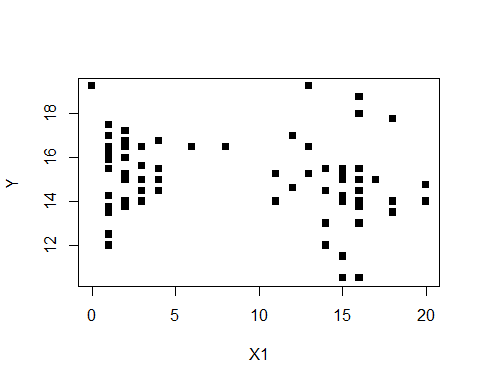
Codes and Plots of question 4 and 5

**Question 4**

# Import input  
loca="C:/Users/Duong Hung/OneDrive - University Of Houston/Msds classes materials/Fall semester/Math 6357-01 Linear models"  
setwd(loca)  
data=read.table("CH06PR18.txt")  
#Rename data  
names(data)=c('Y','X1','X2','X3','X4')  
  
# Plot the scatter plot of all variables. Comment on the plots.  
for (i in 1:4){  
 par(mfrow=c(1,1))  
 plot(data[,i+1],data$Y,xlab = paste('X',i,sep = ""), ylab = 'Y',col='black',pch=14+i)  
}



## b) Fit a regression line to the data

#Build matrix X  
one\_vec=data.frame(c(rep(1,81)))  
names(one\_vec)='1'  
X=data.matrix(cbind(one\_vec,data[,2:5]))  
#Build matrix Y  
Y=data.matrix(data$Y)  
#Calculate beta\_hat  
beta=solve(t(X)%\*%X)%\*%t(X)%\*%Y  
print('vector beta hat is')

## [1] "vector beta hat is"

print(beta)

## [,1]  
## 1 1.220059e+01  
## X1 -1.420336e-01  
## X2 2.820165e-01  
## X3 6.193435e-01  
## X4 7.924302e-06

#Calculate fitted values  
Y\_hat=X%\*%beta  
#Calculate residuals  
res=Y-Y\_hat  
#print(res)  
#Calculate matrix H  
h\_mtx=X%\*%solve(t(X)%\*%X)%\*%t(X)  
i\_mtx=diag(81)  
#residual sum of squares  
sse=t(Y)%\*%(i\_mtx-h\_mtx)%\*%Y  
mse=sse/(81-5)  
s=mse^0.5  
c\_mtx=solve(t(X)%\*%X)  
#Distribution of beta hat  
var\_b=mse[1,1]\*c\_mtx  
#standard error for each beta  
ses=NULL  
for (i in 1:5){  
 ses[i]=sqrt(var\_b[i,i])  
}  
print('standard error for each beta')

## [1] "standard error for each beta"

print(ses)

## [1] 5.779562e-01 2.134261e-02 6.317235e-02 1.086813e+00 1.384775e-06

sig\_lev=0.05  
t\_val=qt(1-sig\_lev/2,81-5)  
ci=array(0,c(5,2))  
for (i in 1:5){  
 ci[i,1]=beta[i]-t\_val\*ses[i]  
 ci[i,2]=beta[i]+t\_val\*ses[i]  
}  
print('Confident interval for each beta')

## [1] "Confident interval for each beta"

print(ci)

## [,1] [,2]  
## [1,] 1.104949e+01 1.335169e+01  
## [2,] -1.845411e-01 -9.952615e-02  
## [3,] 1.561979e-01 4.078352e-01  
## [4,] -1.545232e+00 2.783919e+00  
## [5,] 5.166283e-06 1.068232e-05

for (i in 1:5){print(beta[i]/ses[i])}

[1] 21.10988

[1] -6.654933

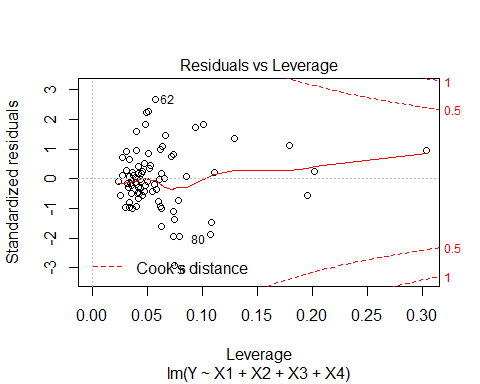
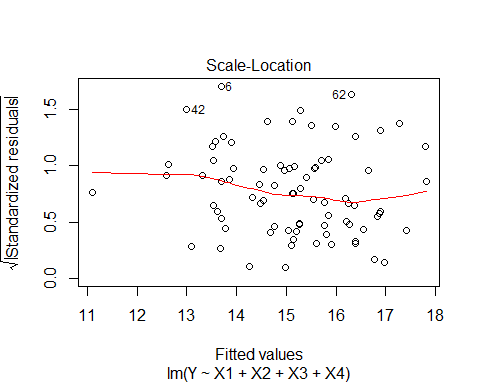
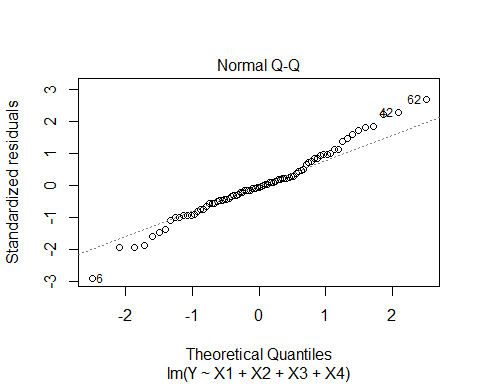
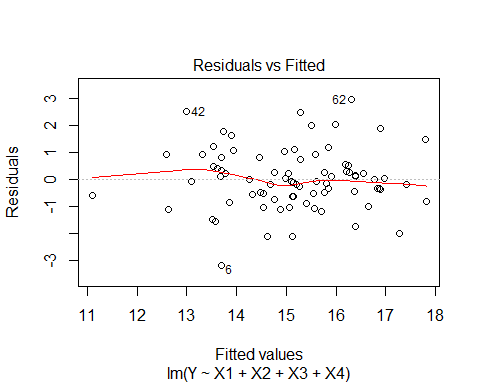
[1] 4.46424

[1] 0.5698714

[1] 5.722446

## c) Conduct the model diagnostic check and conclude with the best model for the given data.

fit=lm(Y~X1+X2+X3+X4,data = data)  
plot(fit)



**crPlots**(fit)

**summary**(fit)

Call:

lm(formula = Y ~ X1 + X2 + X3 + X4, data = data)

Residuals:

Min 1Q Median 3Q Max

-3.1872 -0.5911 -0.0910 0.5579 2.9441

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.220e+01 5.780e-01 21.110 < 2e-16 \*\*\*

X1 -1.420e-01 2.134e-02 -6.655 3.89e-09 \*\*\*

X2 2.820e-01 6.317e-02 4.464 2.75e-05 \*\*\*

X3 6.193e-01 1.087e+00 0.570 0.57

X4 7.924e-06 1.385e-06 5.722 1.98e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.137 on 76 degrees of freedom

Multiple R-squared: 0.5847, Adjusted R-squared: 0.5629

F-statistic: 26.76 on 4 and 76 DF, p-value: 7.272e-14

> fit\_new=lm(Y~X1+X2+X4,data=data)

> summary(fit\_new)

Call:

lm(formula = Y ~ X1 + X2 + X4, data = data)

Residuals:

Min 1Q Median 3Q Max

-3.0620 -0.6437 -0.1013 0.5672 2.9583

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.237e+01 4.928e-01 25.100 < 2e-16 \*\*\*

X1 -1.442e-01 2.092e-02 -6.891 1.33e-09 \*\*\*

X2 2.672e-01 5.729e-02 4.663 1.29e-05 \*\*\*

X4 8.178e-06 1.305e-06 6.265 1.97e-08 \*\*\*

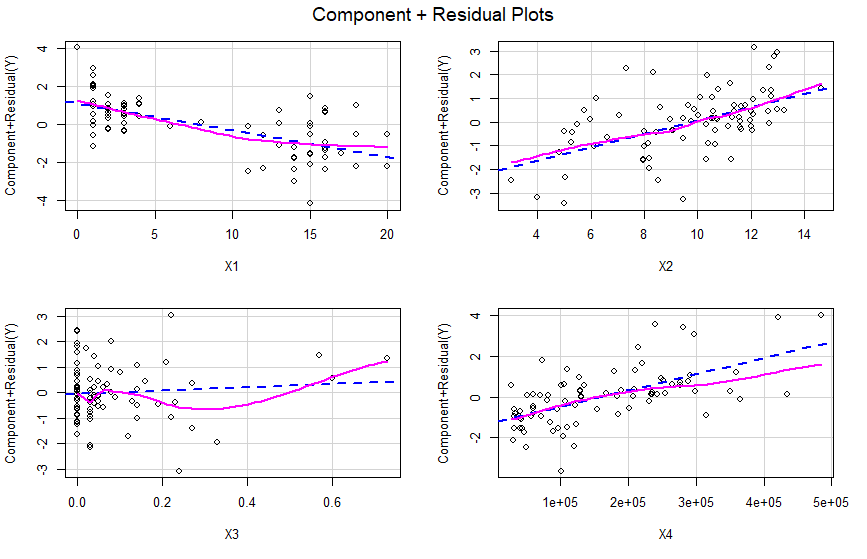
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.132 on 77 degrees of freedom

Multiple R-squared: 0.583, Adjusted R-squared: 0.5667

F-statistic: 35.88 on 3 and 77 DF, p-value: 1.295e-14



**Question 5: Calculate the confidence intervals and prediction intervals of 95% for the 3 properties with the given feature**.

x\_0=c(1,1,1)  
x\_1=c(4,6,12)  
x\_2=c(10,11.5,12.5)  
x\_3=c(0.1,0,0.32)  
x\_4=c(80000,120000,340000)  
ndata=data.frame(X1=x\_1,X2=x\_2,X3=x\_3,X4=x\_4)  
predict(fit,ndata,interval = 'confidence',level = 0.95)

## fit lwr upr  
## 1 15.14850 14.76829 15.52870  
## 2 15.54249 15.15366 15.93132  
## 3 16.91384 16.18358 17.64410

predict(fit,ndata,interval = 'prediction',level = 0.95)

## fit lwr upr  
## 1 15.14850 12.85249 17.44450  
## 2 15.54249 13.24504 17.83994  
## 3 16.91384 14.53469 19.29299