Assignment 3

Sohaib Syed

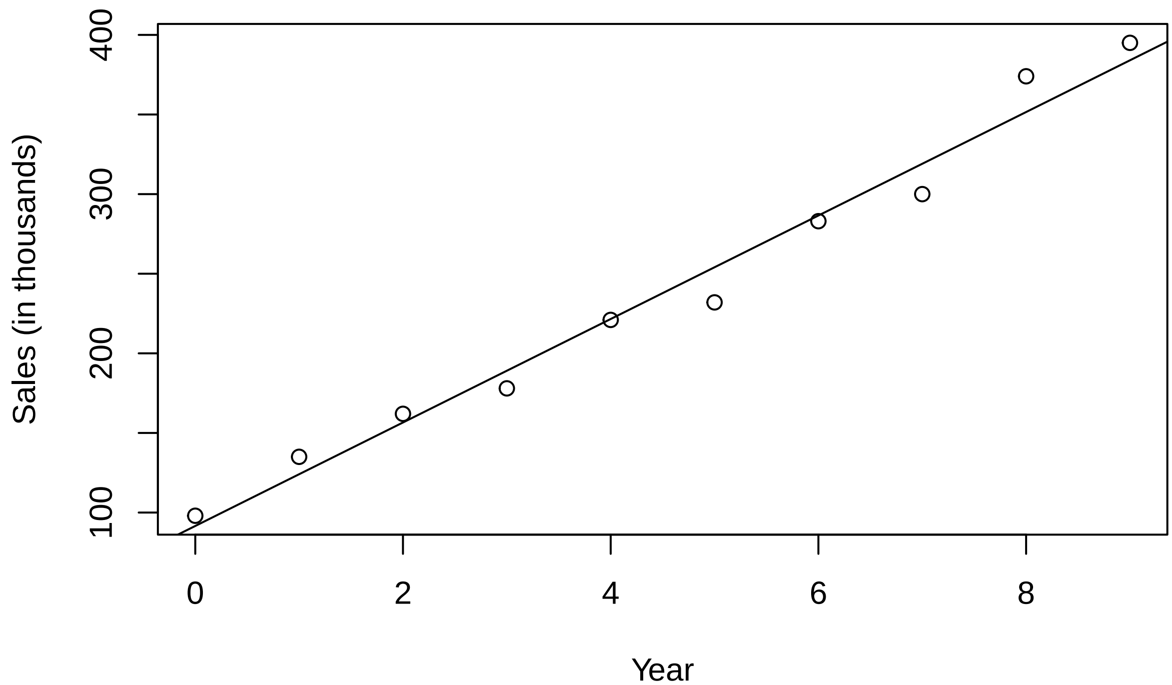
2022-10-02

Contents

Problem 1

a

```
sales_data<-read.table('CH03PR17.txt',col.names = c('y','x'))
par(mfrow=c(1,1))
plot(sales_data$x,sales_data$y,xlab = 'Year',ylab='Sales (in thousands)')
f<-lm(sales_data$y~sales_data$x)
abline(f)
```



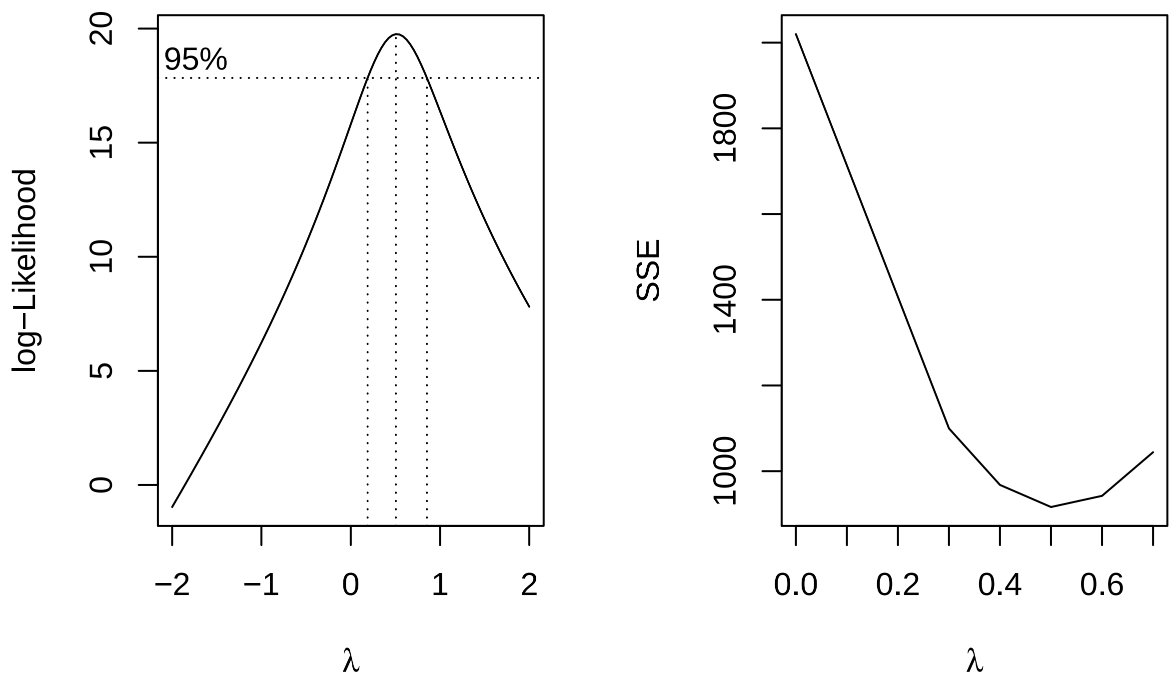

```
summary(f)
```

```
##
## Call:
## lm(formula = sales_data$y ~ sales_data$x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.049  -9.177   2.446   9.814  22.461
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    91.564      8.814   10.39 6.38e-06 ***
## sales_data$x   32.497      1.651   19.68 4.62e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15 on 8 degrees of freedom
## Multiple R-squared:  0.9798, Adjusted R-squared:  0.9772
## F-statistic: 387.4 on 1 and 8 DF,  p-value: 4.62e-08
```

A linear function may not be adequate here because as year increase the rate of sales increases non-linearly.

b

```
par(mfrow=c(1,2))
boxcox(y~x,data=sales_data)
boxcox.sse(sales_data$x,sales_data$y,l=seq(.3,.7,.1))
```




```
##   lambda       SSE
## 6    0.0 2019.8767
## 1    0.3 1099.7093
## 2    0.4  967.9088
## 3    0.5  916.4048
## 4    0.6  942.4498
## 5    0.7 1044.2384
```

It is suggested to use Y^.5 aka sqrt(Y)

c

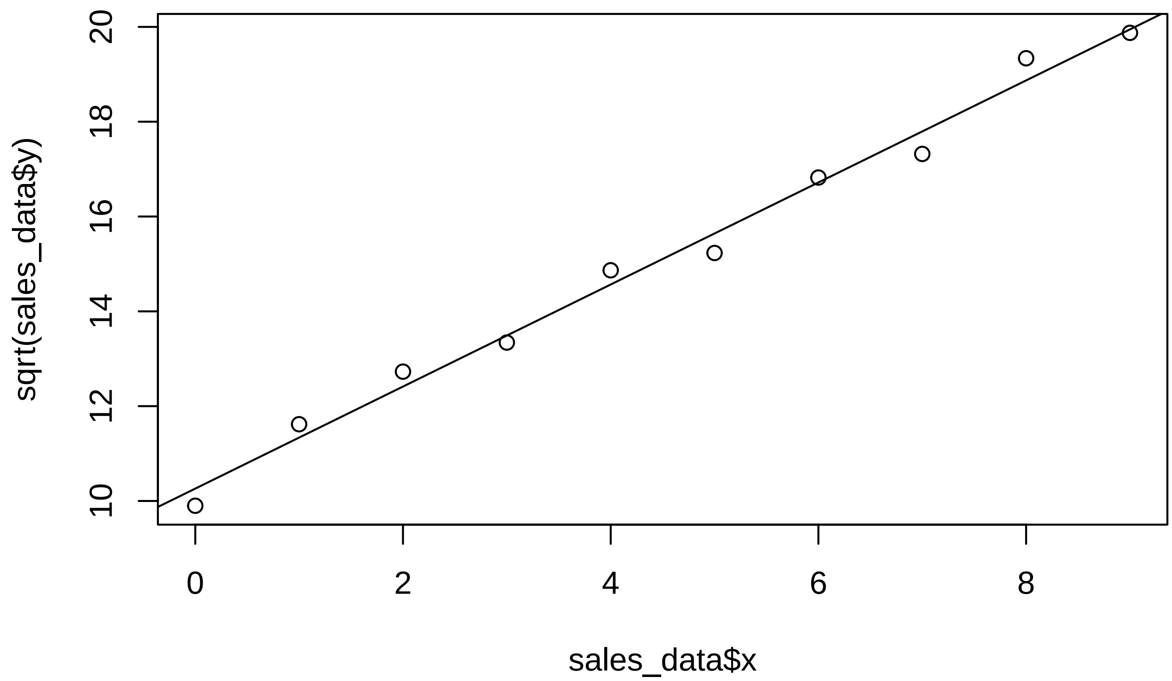
```
f1<-lm(sqrt(sales_data$y)~sales_data$x)

summary(f1)
```

```
##
## Call:
## lm(formula = sqrt(sales_data$y) ~ sales_data$x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47447 -0.30811  0.01549  0.29541  0.46781
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.26093    0.21290   48.20 3.80e-11 ***
## sales_data$x  1.07629    0.03988   26.99 3.83e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3622 on 8 degrees of freedom
## Multiple R-squared:  0.9891, Adjusted R-squared:  0.9878
## F-statistic: 728.4 on 1 and 8 DF,  p-value: 3.826e-09
```

sqrt(Y)=10.26093+1.07626x

d

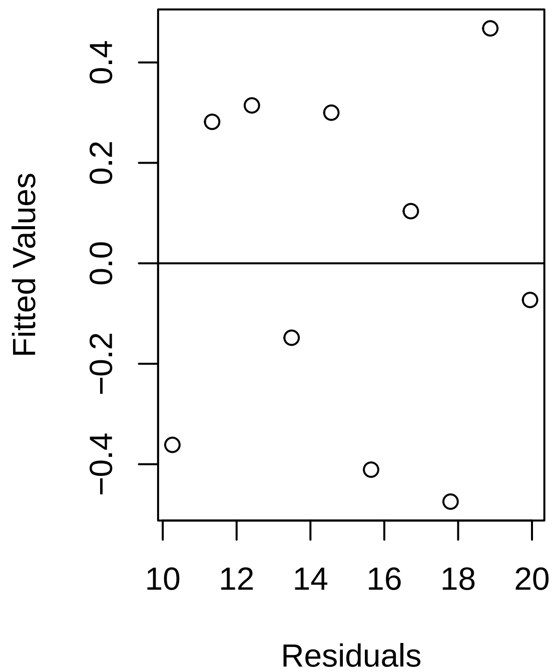
```
plot(sales_data$x,sqrt(sales_data$y))
abline(f1)
```

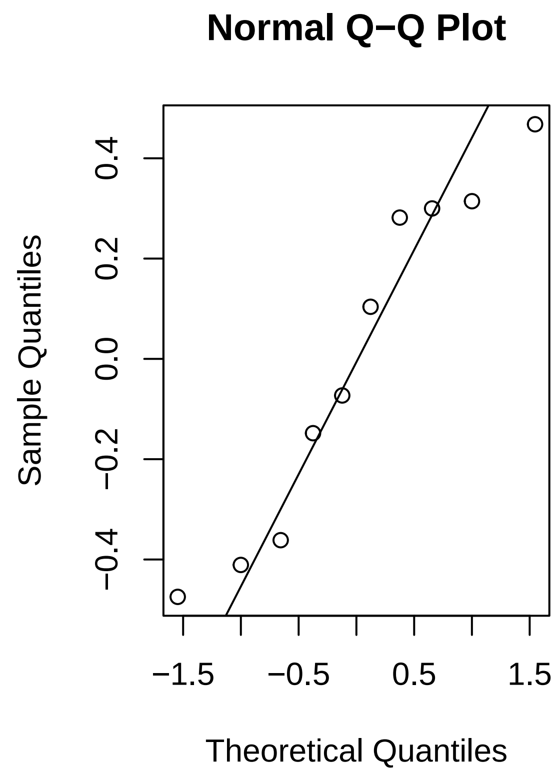


Yes the line is a great fit on the data.

e

```
ei<-f1$residuals
yhat<-f1$fitted.values
par(mfcol=c(1,2))
plot(yhat,ei,xlab='Residuals',ylab='Fitted Values')
abline(0,0)
qqnorm(residuals(f1))
qqline(residuals(f1))
```





Plots show errors are there are no pattern to residuals and the residuals are approximately normally distributed as they are close to the line

f

sqrt(Sales in thousands)=10.26093+1.07626(coded year)

Problem 2

##a

```
mass_data<-read.table('CH01PR27.txt',col.names = c('y','x'))
xh<-c(45,55,65)
f2<-lm(y~x,data=mass_data)
f2_sum<-summary(f2)
anova(f2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## x          1 11627.5 11627.5  174.06 < 2.2e-16 ***
## Residuals 58  3874.4    66.8
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
pred<-predict(f2,newdata = data.frame(x=xh),se.fit = T,level=.95)
W <-sqrt( 2 * qf(0.95, 2, 58) )
f2_sum
```

```
##
## Call:
## lm(formula = y ~ x, data = mass_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1368  -6.1968  -0.5969   6.7607  23.4731
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 156.3466     5.5123   28.36   <2e-16 ***
## x            -1.1900     0.0902  -13.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.173 on 58 degrees of freedom
## Multiple R-squared:  0.7501, Adjusted R-squared:  0.7458
## F-statistic: 174.1 on 1 and 58 DF,  p-value: < 2.2e-16
```

```r
lower<-pred$fit-W * pred$se.fit
upper<-pred$fit+W * pred$se.fit
lower
```

```
##        1        2        3
## 98.48916 88.01540 76.11248
```

```r
upper
```

```
##         1         2         3
## 107.10437  93.77822  81.88123
```

Xh=45; 98.489 <=E{yh}<= 107.104

Xh=55; 88.015 <=E{yh}<= 93.778

Xh=65; 76.113 <=E{yh}<= 81.881

b

The WH procedure is better for larger g, so no, not for this problem. It's not the most efficient.

c

```r
xhbef=c(48,59,74)
Wbef<-qt(1-.05/6,58)
pred2<-predict(f2,newdata = data.frame(x=xhbef),se.fit = T,level=.95)
Sxx <- sum( mass_data$x * mass_data$x) - length(mass_data$x) * (mean(mass_data$x))^2
varR <- (f2_sum$sigma)^2
SE <- sqrt(varR*((1/length(mass_data$x) + (xhbef - mean(mass_data$x))^2/Sxx) + 1))
pred2$fit+Wbef*SE
```

```
##         1         2         3
## 119.71815 106.45537  88.84195
```

```r
pred2$fit-Wbef*SE
```

```
##        1        2        3
## 78.73541 65.81829 47.73184
```

Xh=48; 78.73541 <=E{yh}<= 119.71815

Xh=59; 65.81829 <=E{yh}<= 106.45537

Xh=74; 47.73184 <=E{yh}<= 88.84195

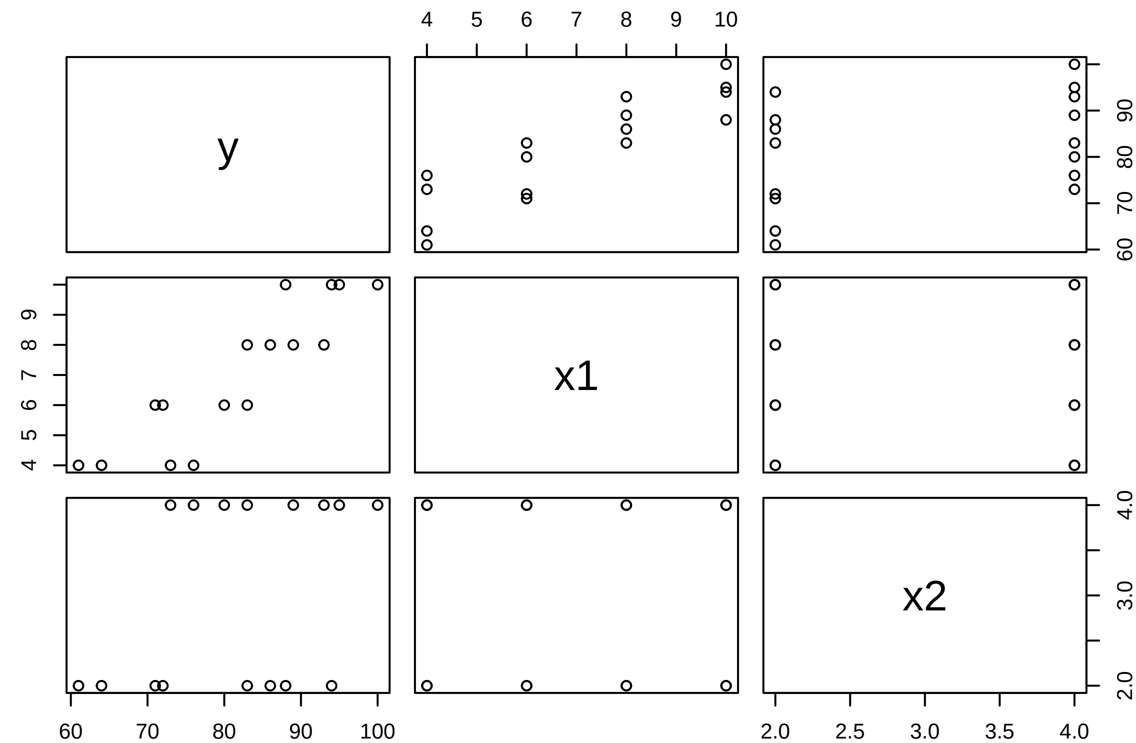
d

Yes, the three prediction intervals will need to be recalculated. Same for the Scheffe Procedure.

Problem 3

a

```r
brand_data<-read.table('CH06PR05.txt',col.names = c('y','x1','x2'))
pairs(~y+x1+x2,data=brand_data)
```




```
cor(brand_data)
```

```
##            y        x1        x2
## y  1.0000000 0.8923929 0.3945807
## x1 0.8923929 1.0000000 0.0000000
## x2 0.3945807 0.0000000 1.0000000
```

The scatter plot shows general relationship between Y and input variables and the correlation matrix shows that moisture(x1) has a very strong positive correlation with brand liking(y).

b

```
fit3<-lm(y~x1+x2,data = brand_data)
summary(fit3)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = brand_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.400 -1.762  0.025  1.587  4.200
```

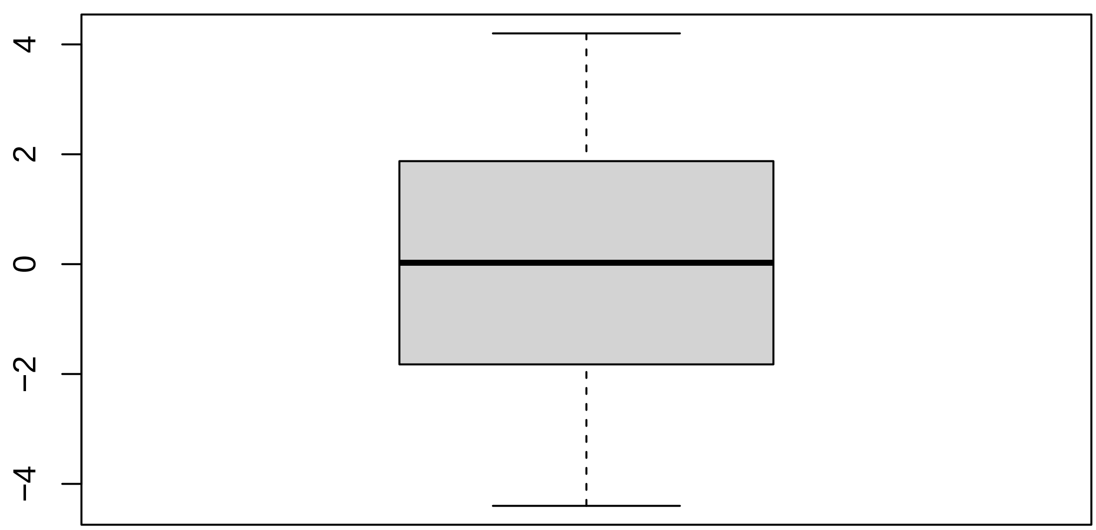
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.6500     2.9961  12.566 1.20e-08 ***
## x1            4.4250     0.3011  14.695 1.78e-09 ***
## x2            4.3750     0.6733   6.498 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.693 on 13 degrees of freedom
## Multiple R-squared:  0.9521, Adjusted R-squared:  0.9447
## F-statistic: 129.1 on 2 and 13 DF,  p-value: 2.658e-09
```

Y= 4.425x1+4.375x2+37.65 is the regression function.

B1 is how much the moisture content affects the brand liking.

c

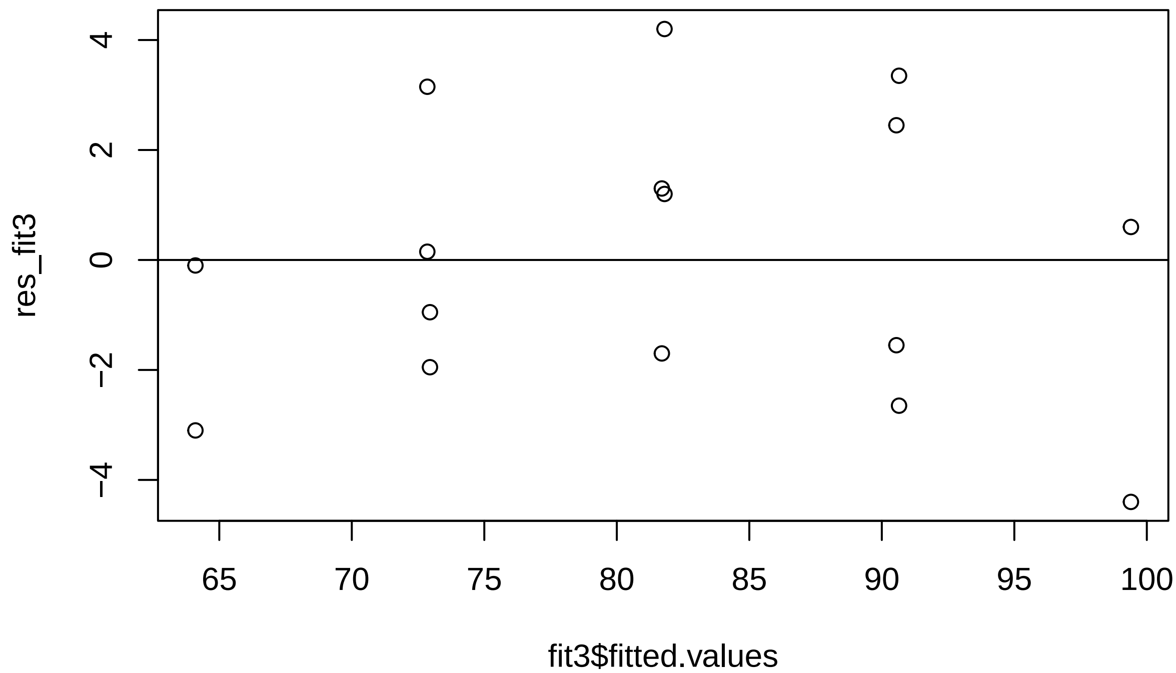
```r
res_fit3<-residuals(fit3)
boxplot(res_fit3)
```



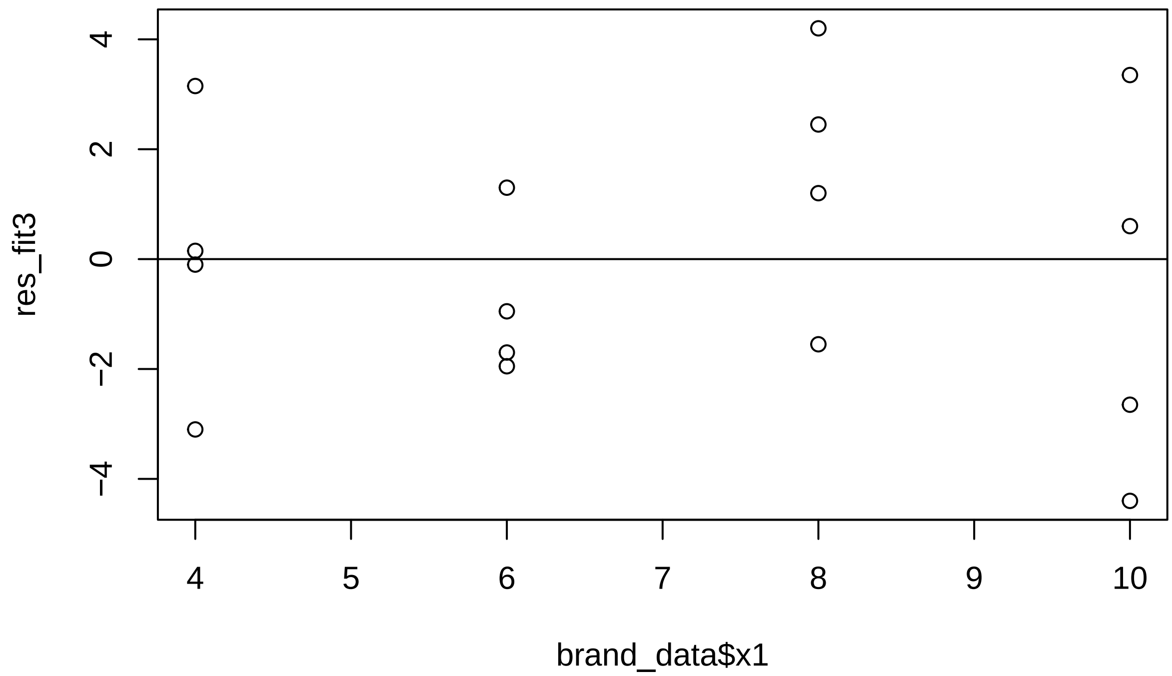
The boxplot shows the spread of the residuals and their quantiles. There seems to be no outliers and the boxplot is symmetrical.

d

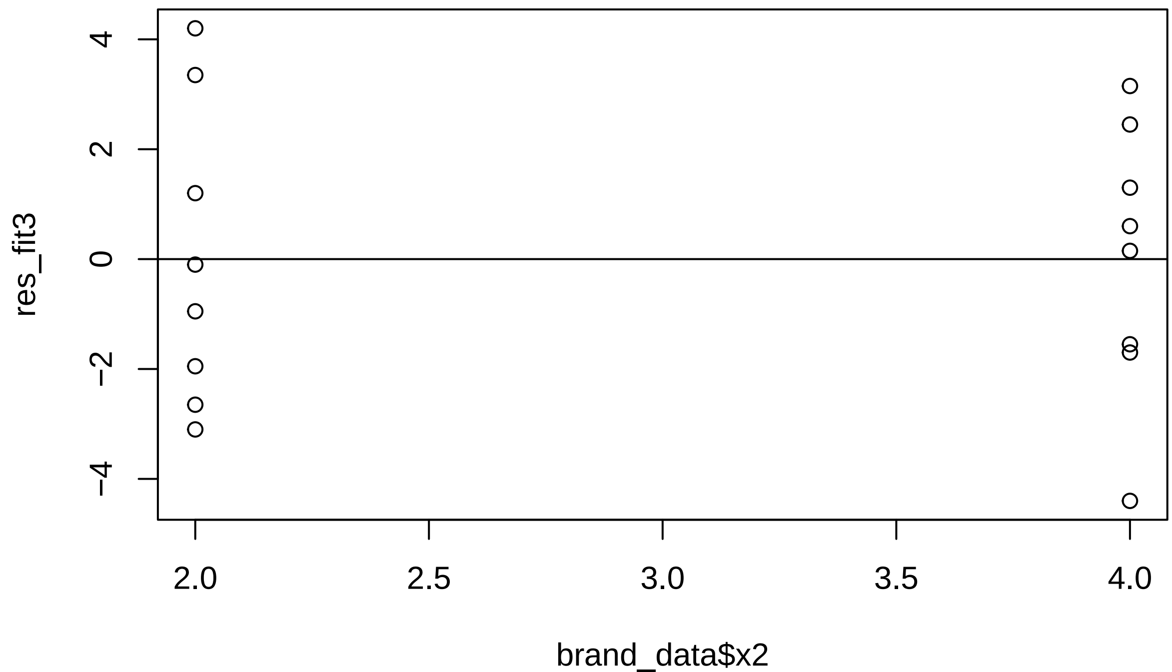
```r
plot(fit3$fitted.values,res_fit3)
abline(0,0)
```



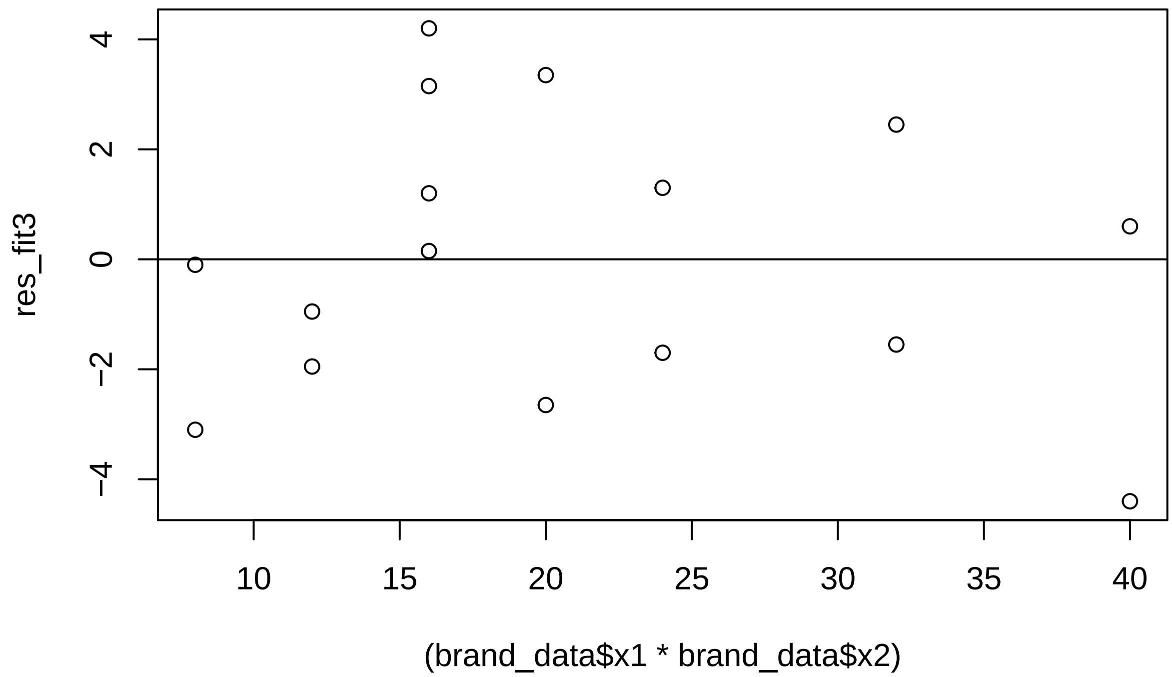

```r
plot(brand_data$x1,res_fit3)
abline(0,0)
```



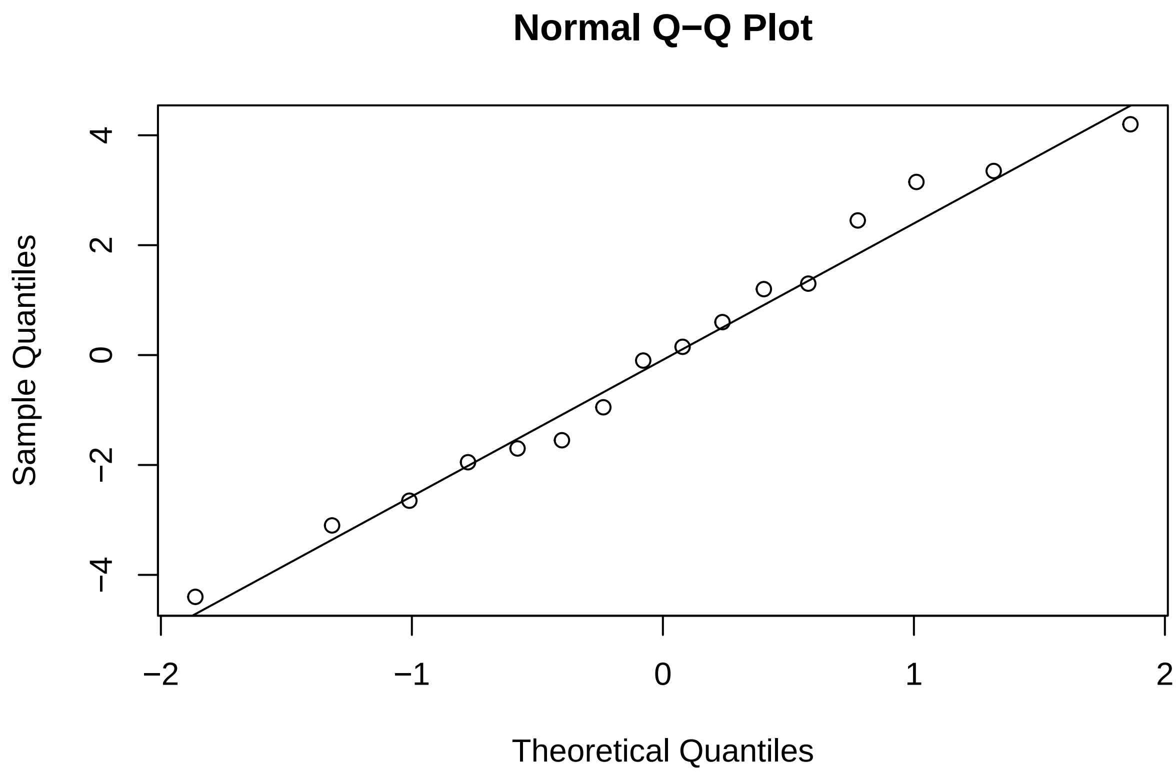

```
plot(brand_data$x2,res_fit3)
abline(0,0)
```




```
plot((brand_data$x1*brand_data$x2),res_fit3)
abline(0,0)
```




```
qqnorm(res_fit3)
qqline(res_fit3)
```



The residuals do not seem random and there are repeated values. The normal plot however shows the residuals follow close to linear line.

e

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lm(res_fit3~brand_data$x1)
```

```
##
## Call:
## lm(formula = res_fit3 ~ brand_data$x1)
##
```

```
## Coefficients:
##   (Intercept)  brand_data$x1
##    -3.014e-16      3.413e-17
```

```
bptest(fit3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit3
## BP = 2.0441, df = 2, p-value = 0.3599
```

```
chi_val=qchisq(.99, df=2)
chi_val
```

```
## [1] 9.21034
```

```
# Alternatives
# H0:y1=0  and Ha:y1 != 0

# Decision Rule: if X^2BP < chi-square distribution
```

Conclude that y1=0 since 2.0441< 9.21

f

```
alpha=.01
fit_lack<-lm(y~as.factor(x1)+as.factor(x2),data = brand_data)
lackfit<-anova(fit3,fit_lack)
#Alternatives: H0:E{Y} = b0 + b1x1 + b1x2 and Ha: E{Y} != b0 + b1x1 + b1x2
# Reject H0 if F-ratio > F a,m-p,n-m
(lackfit$F)<qf(1-alpha,5,8)
```

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

Conclude: accept H0 since Fratio not bigger than F distribution

Problem 4

a

```
commercial_data<-read.table("CH06PR18.txt",col.names = c('y','x1','x2','x3','x4'))
stem(commercial_data$x1)
```

```
##
##   The decimal point is at the |
##
##    0 | 0000000000000000
##    2 | 00000000000000000000000
##    4 | 00000
##    6 | 0
##    8 | 0
##   10 | 00
##   12 | 00000
##   14 | 0000000000000
##   16 | 0000000000
##   18 | 000
##   20 | 00
```

```
stem(commercial_data$x2)
```

```
##
##   The decimal point is at the |
##
##    2 | 0
##    4 | 080003358
##    6 | 012613
##    8 | 00001223456001555689
##   10 | 01334456667777812334466668
##   12 | 00011115777889002
##   14 | 6
```

```
stem(commercial_data$x3)
```

```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   0 | 00000000000000000000000000000233333333333444444555555556678889
##   1 | 023444469
##   2 | 1223477
##   3 | 3
##   4 |
##   5 | 7
##   6 | 0
##   7 | 3
```

```
stem(commercial_data$x4)
```

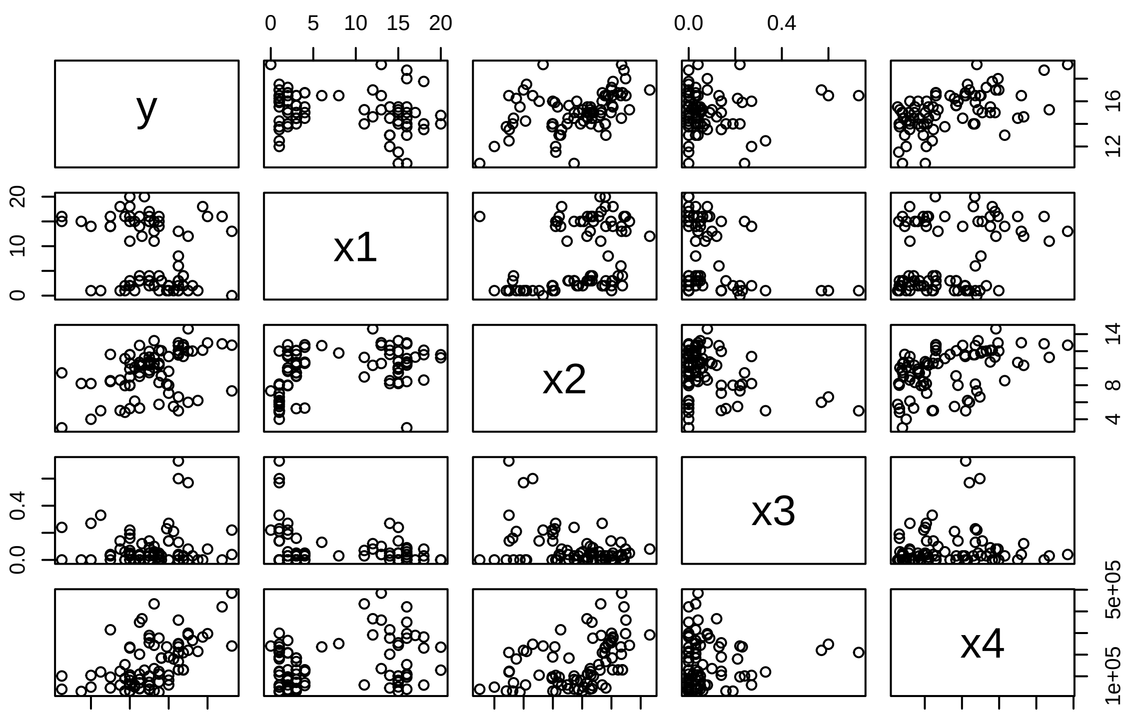
```
##
##   The decimal point is 5 digit(s) to the right of the |
##
##   0 | 333333444444
##   0 | 555666667778899
##   1 | 000001111222333334
##   1 | 578889
##   2 | 011122334444
```

```
##    2 | 555788899
##    3 | 002
##    3 | 567
##    4 | 23
##    4 | 8
```

The stem and leaf plots shows the frequency at which certain classes of values occur

b

```r
pairs(~y+.,data=commercial_data)
```




```r
cor(commercial_data)
```

```
##              y         x1         x2          x3         x4
## y   1.00000000 -0.2502846  0.4137872  0.06652647 0.53526237
## x1 -0.25028456  1.0000000  0.3888264 -0.25266347 0.28858350
## x2  0.41378716  0.3888264  1.0000000 -0.37976174 0.44069713
## x3  0.06652647 -0.2526635 -0.3797617  1.00000000 0.08061073
## x4  0.53526237  0.2885835  0.4406971  0.08061073 1.00000000
```

The rate(y) is strongly correlated to two predictors: first square footage(x4) and then expenses(x2). expenses(x2) and square footage(x4) is are also correlated which explains why both of those predictors also have strong correlation with rate(y). It also makes sense to see expenses(x2) and vacancy(x3) have strong negative correlation, as if more places are vacant there are less expenses.

c

```
commercialfit<-lm(y~.,data=commercial_data)
sum_comm<-summary(commercialfit)
sum_comm
```

```
##
## Call:
## lm(formula = y ~ ., data = commercial_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
```

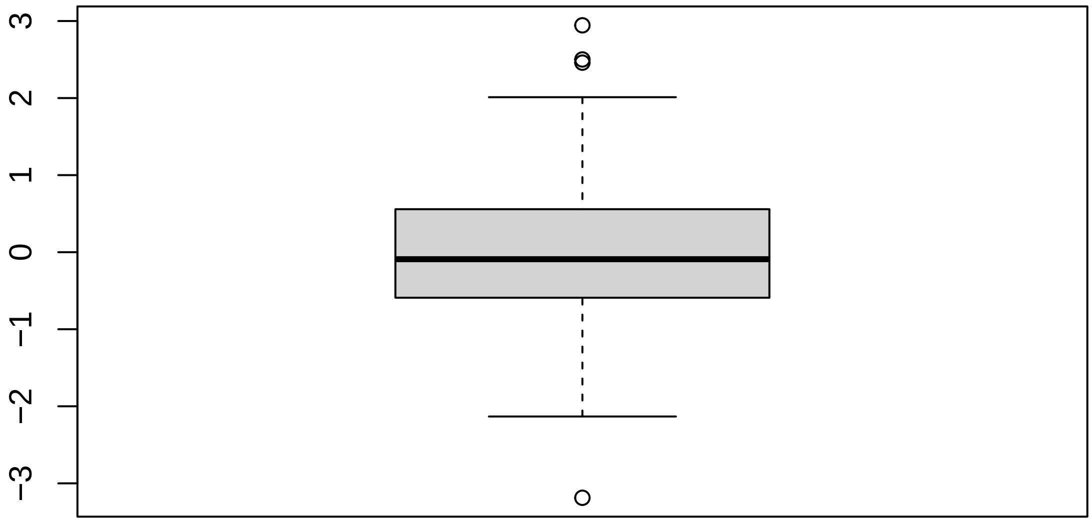
```
anova(commercialfit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 14.819  14.819 11.4649  0.001125 **
## x2         1 72.802  72.802 56.3262 9.699e-11 ***
## x3         1  8.381   8.381  6.4846  0.012904 *
## x4         1 42.325  42.325 32.7464 1.976e-07 ***
## Residuals 76 98.231   1.293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Y=12.2-.1420x1+.2820x2+.6193x3+.000007924x4

d

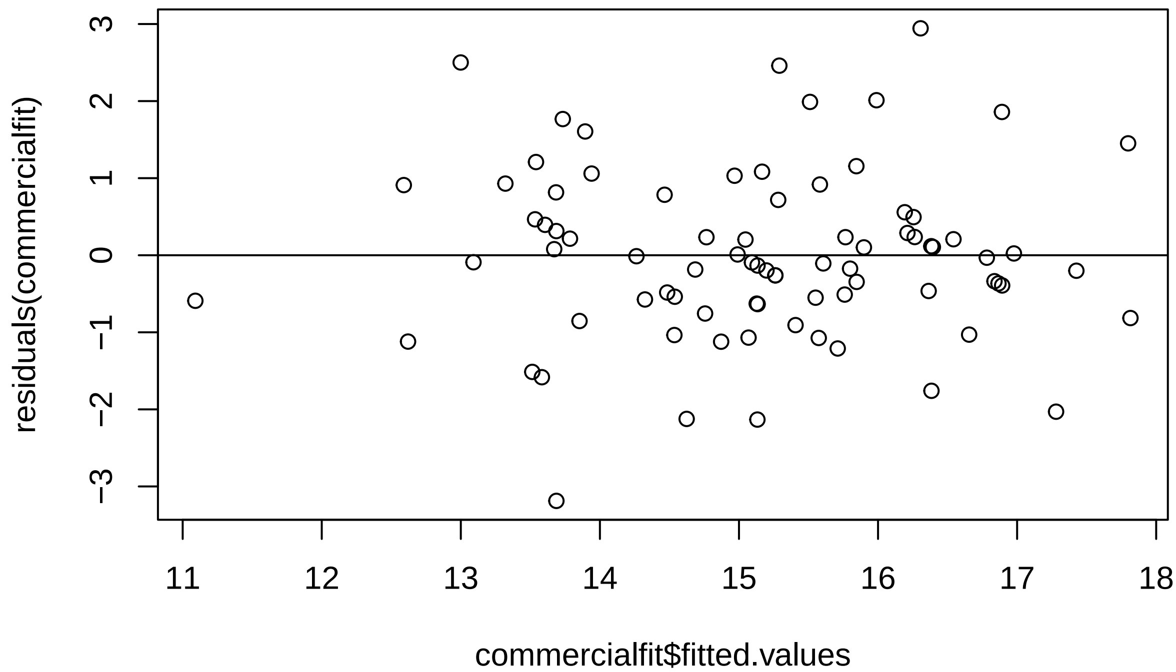
```
boxplot(resid(commercialfit))
```



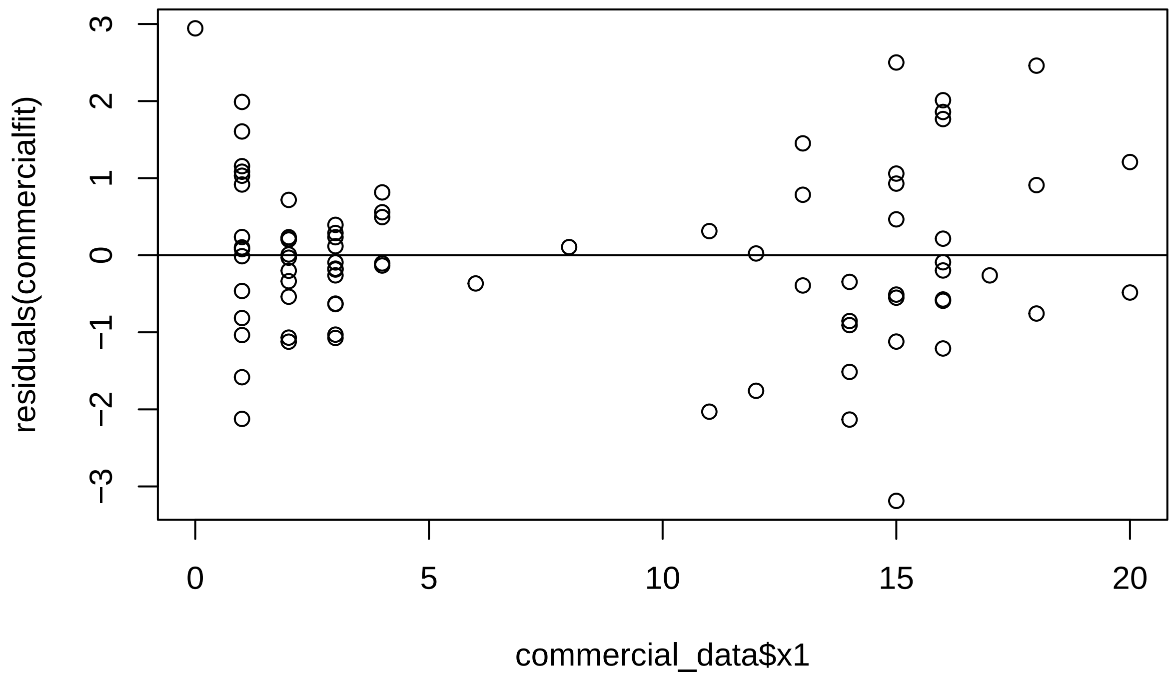
No. it seems like there are a number of outliers outside of the boxplot, especially at the top. If there were no outliers however the boxplot would be symmetrical.

e

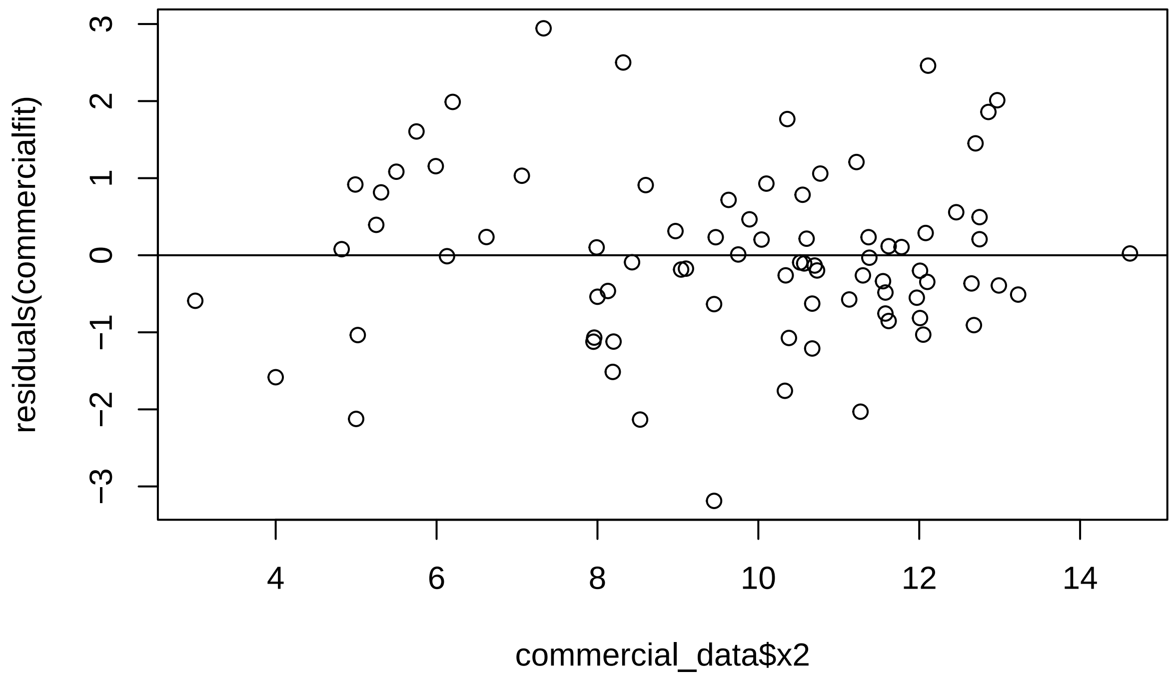
```
plot(commercialfit$fitted.values,residuals(commercialfit))
abline(0,0)
```



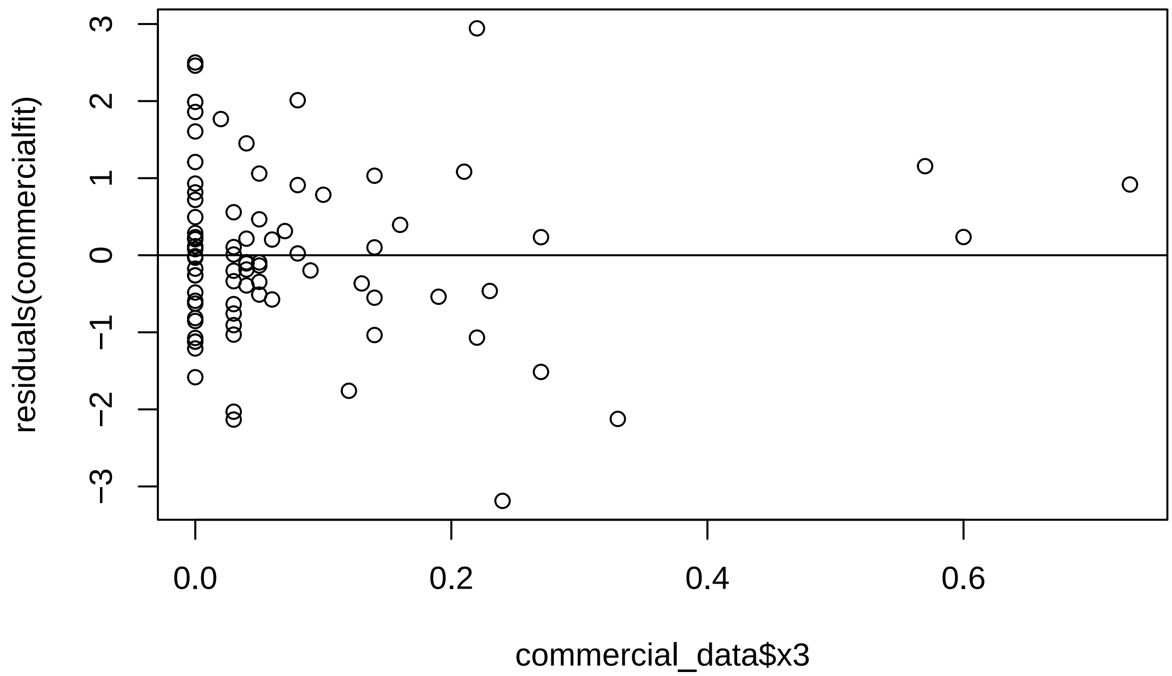

```
plot(commercial_data$x1,residuals(commercialfit))
abline(0,0)
```



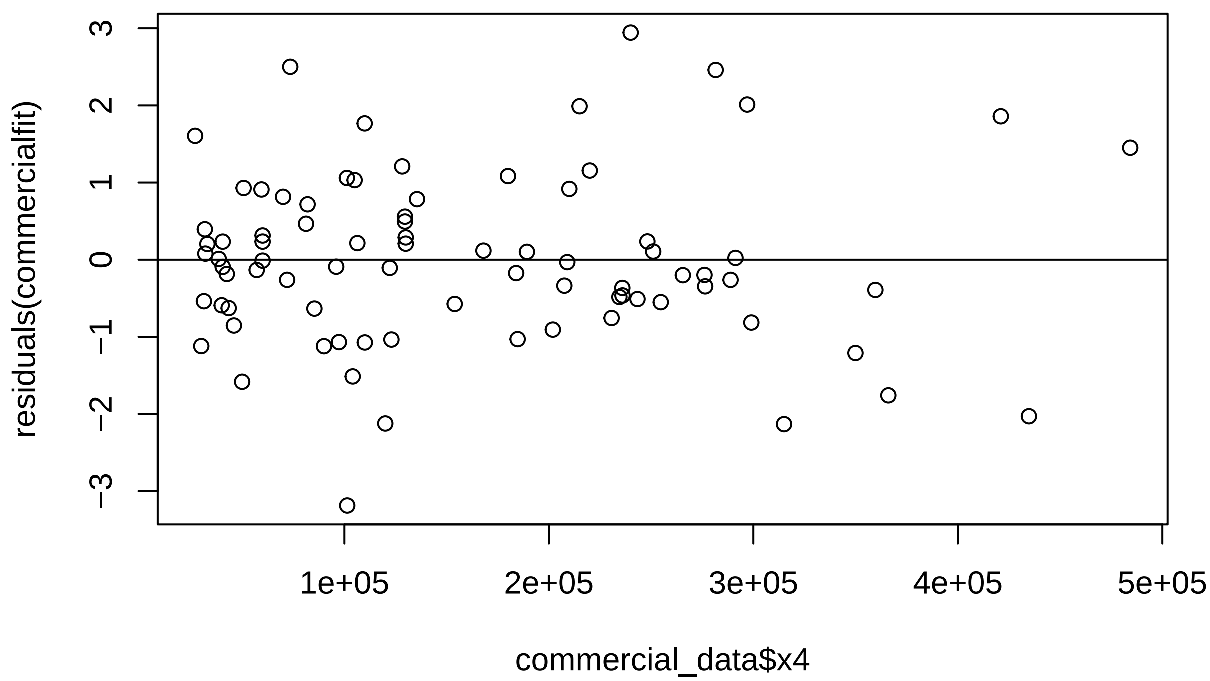

```
plot(commercial_data$x2,residuals(commercialfit))
abline(0,0)
```



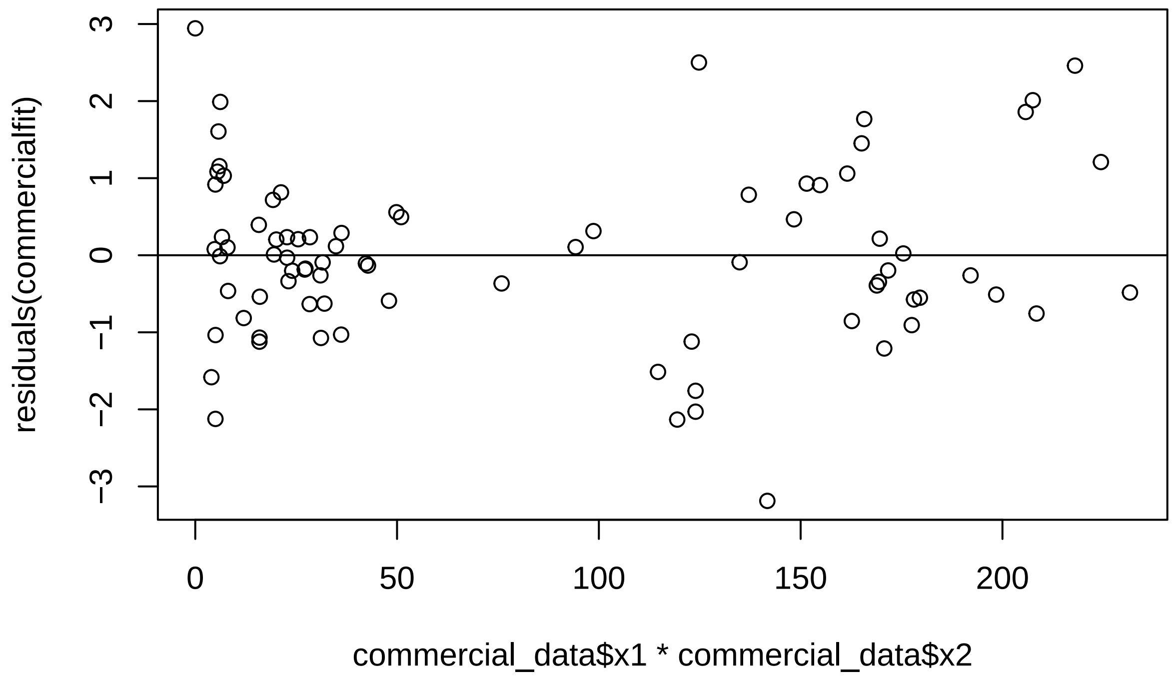

```
plot(commercial_data$x3,residuals(commercialfit))
abline(0,0)
```



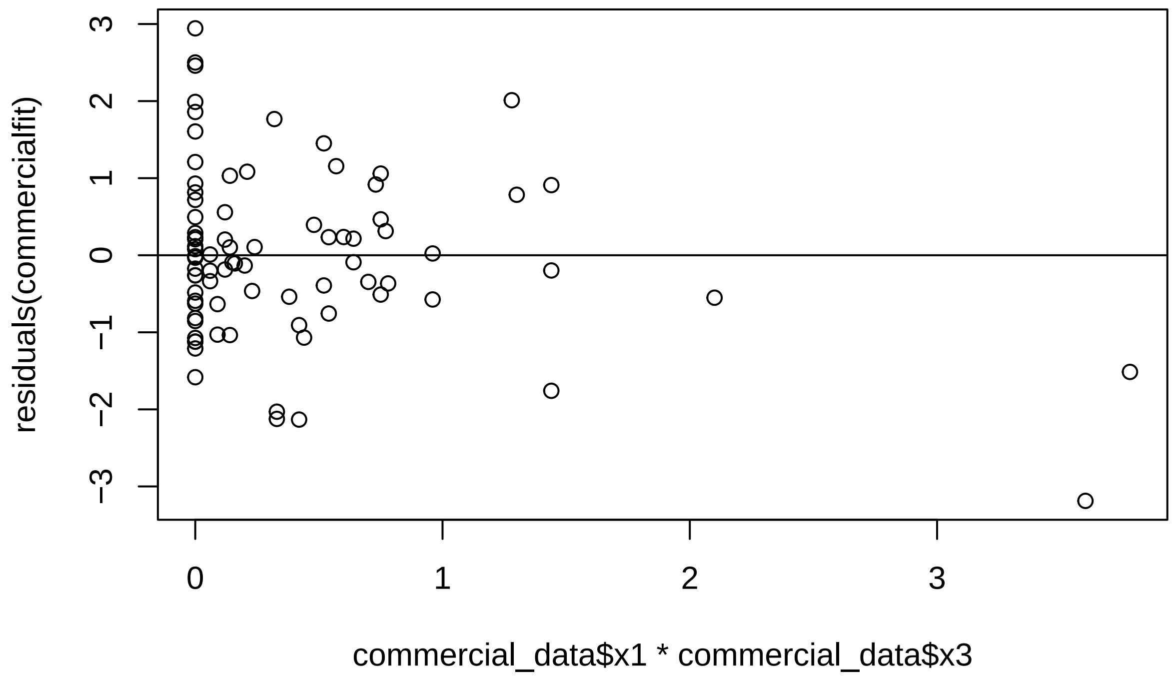

```
plot(commercial_data$x4,residuals(commercialfit))
abline(0,0)
```



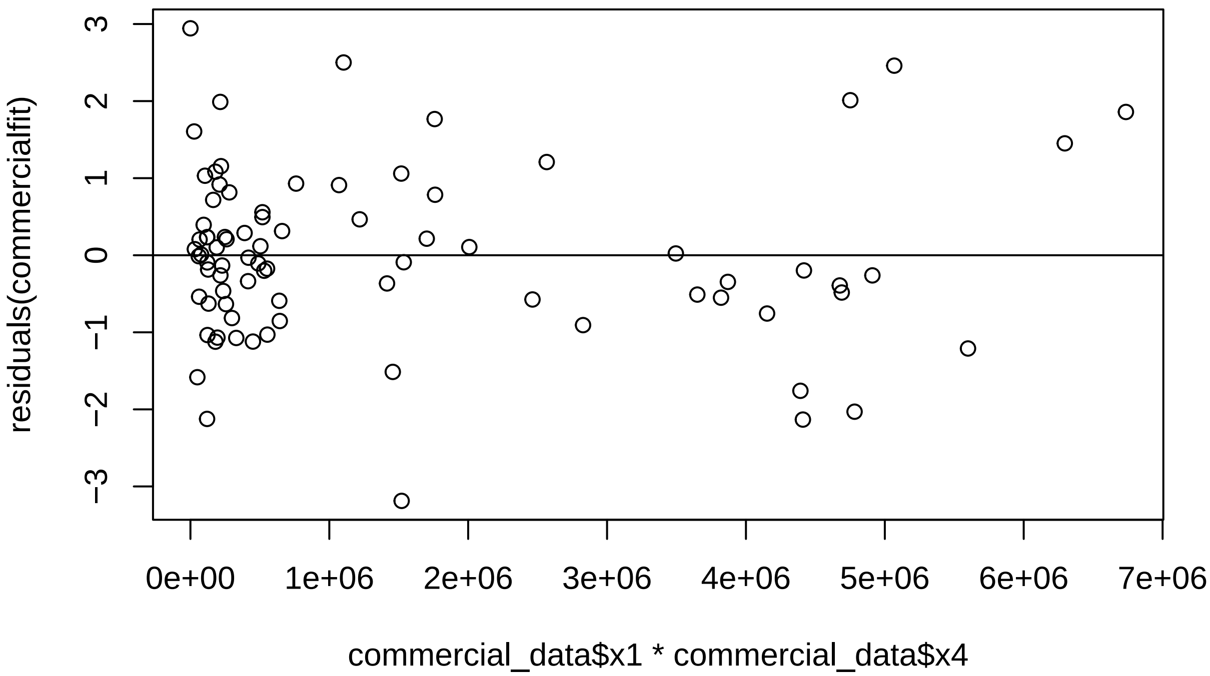

```
plot(commercial_data$x1*commercial_data$x2,residuals(commercialfit))
abline(0,0)
```



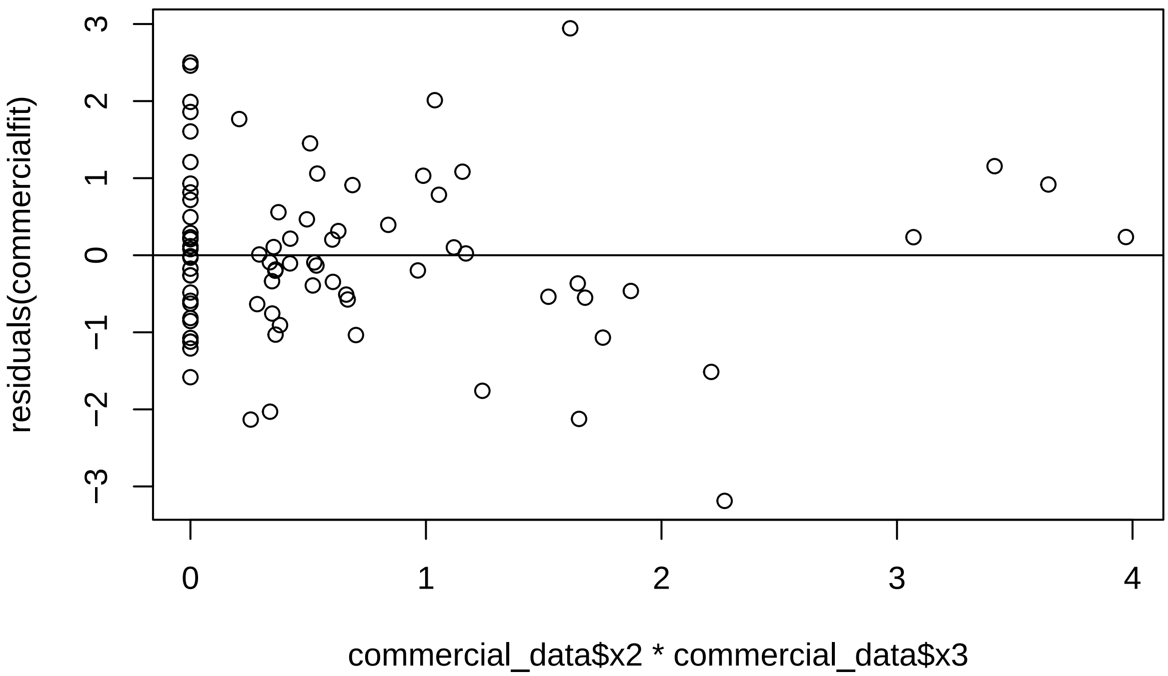

```
plot(commercial_data$x1*commercial_data$x3,residuals(commercialfit))
abline(0,0)
```



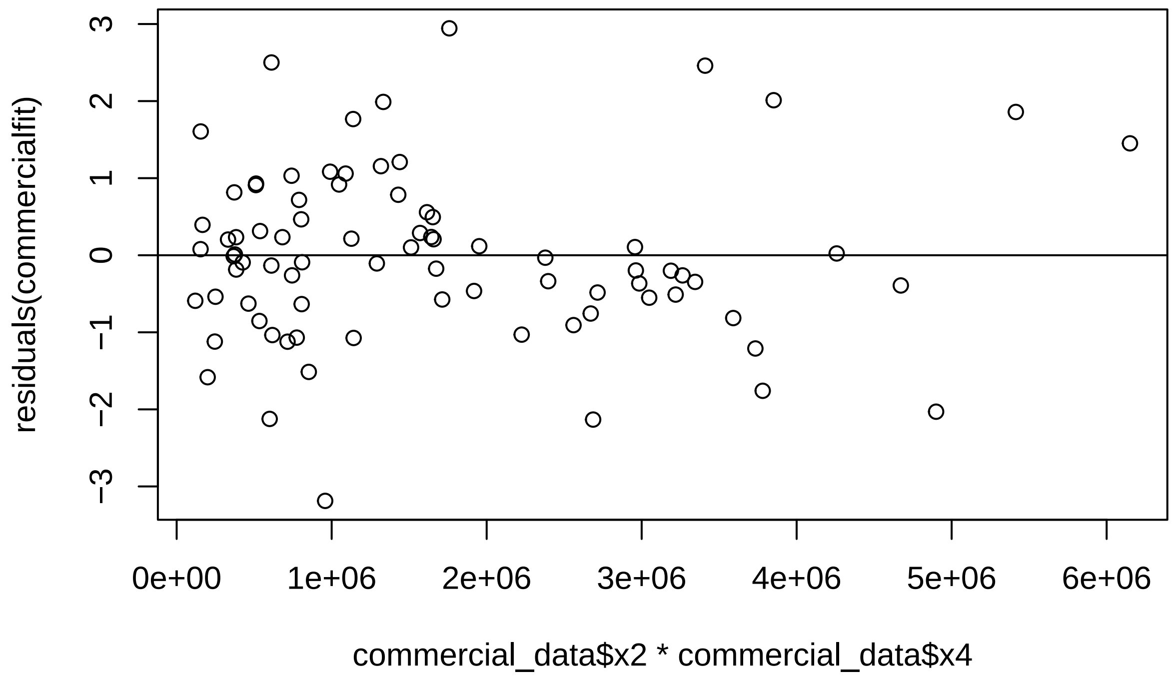

```
plot(commercial_data$x1*commercial_data$x4,residuals(commercialfit))
abline(0,0)
```



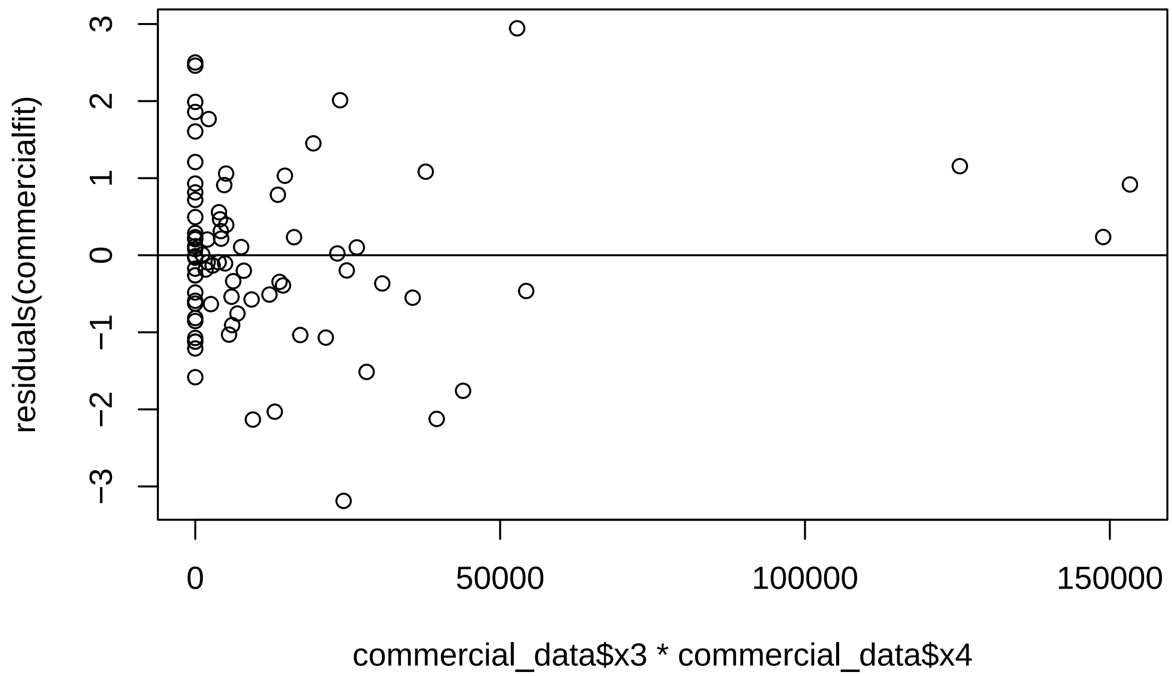

```
plot(commercial_data$x2*commercial_data$x3,residuals(commercialfit))
abline(0,0)
```



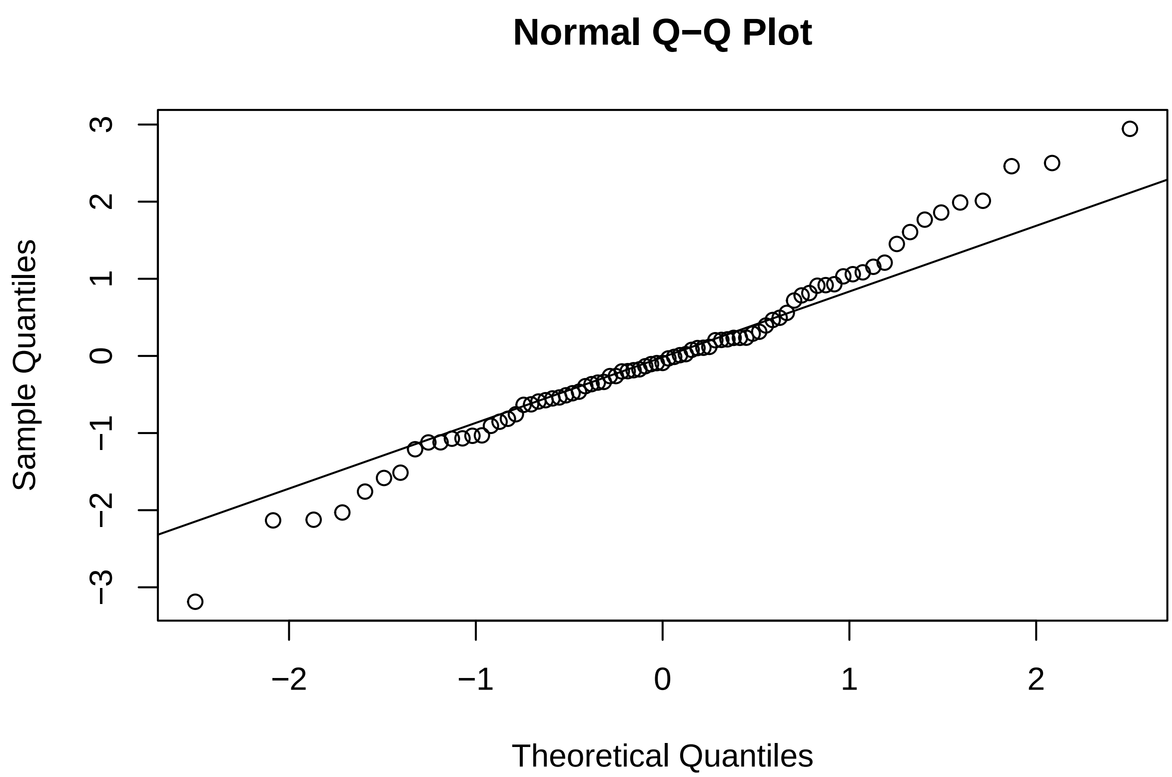

```
plot(commercial_data$x2*commercial_data$x4,residuals(commercialfit))
abline(0,0)
```




```
plot(commercial_data$x3*commercial_data$x4,residuals(commercialfit))
abline(0,0)
```




```
qqnorm(residuals(commercialfit))
qqline(residuals(commercialfit))
```



The residual plots show that the fitted values are apprpriate for a linear fit, also the x2,x4 and x2*x4, terms are also good terms to be used in a linear model since the residual points are mostly random. The rest of the plots show some pattern or at least they aren't showing randomness. The normal plot also seems to not completely follow a linear line as the points at the ends start to increase their distance from the line. the outliers may be occuring because of the non-randomness of x1 and x3.

f

No, because each xi would need to have repeating Y values which doesn't occur with the given data.

g

Decision rule with alpha=0.5, if |t*BF| <= t alpha/2,n-2 then error variance is constant

```
rownum_of_ordered_fitted<-order(commercialfit$fitted.values)
fortysmallest_fitted<-commercial_data[rownum_of_ordered_fitted[1:40],]
restof_fitted<-commercial_data[rownum_of_ordered_fitted[41:81],]

group1_fitted<-lm(y~.,data=fortysmallest_fitted)
group2_fitted<-lm(y~.,data=restof_fitted)

d1<-abs(residuals(group1_fitted)-median(residuals(group1_fitted)))
mean(d1)
```

```
## [1] 0.6963691
```

```
d2<-abs(residuals(group2_fitted)-median(residuals(group2_fitted)))
mean(d2)
```

```
## [1] 0.7492604
```

```
sdd1<-sum((d1-mean(d1))^2)
sdd2<-sum((d2-mean(d2))^2)
s_for_comm<-sqrt((sdd1+sdd2)/79)

t_star_comm<-(mean(d1)-mean(d2))/(s_for_comm*sqrt((1/40)+(1/41)))

t_star_comm<qt(1-.05/2,79)
```

```
## [1] TRUE
```

conclusion: Error variance is constant

Problem 5

a

alternatives

H0: b1 = b2 = b3 = b4 = 0, Ha: not all bk in H0 equal 0

Decision rule:

using F-Ratio; if F*>F a,p-q,n-p then reject H0

```
comm_anova<-anova(commercialfit)
sum_comm
```

```
##
## Call:
## lm(formula = y ~ ., data = commercial_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
```

```
comm_MSR<-mean(comm_anova$`Mean Sq`[1:4])
comm_MSE<-comm_anova$`Mean Sq`[5]
F_star_comm<-comm_MSR/comm_MSE
comm_F_score<-qf(1-.05,4,76)
F_star_comm>comm_F_score
```

```
## [1] TRUE
```

Conclusion

Reject H0. This means that there is atleast one Bk that is influencial to the data. The p-value of the test is 2.272e-14, almost 0.

b

```
commercial_upperbounds_betas<-sum_comm$coefficients[2:5,'Estimate']+qt(1-0.05/8,76)*sum_comm$coefficien
commercial_lowerbounds_betas<-sum_comm$coefficients[2:5,'Estimate']-qt(1-0.05/8,76)*sum_comm$coefficien
commercial_upperbounds_betas
```

```
##            x1            x2            x3            x4
## -8.742769e-02  4.436456e-01  3.399999e+00  1.146731e-05
```

```
commercial_lowerbounds_betas
```

```
##            x1            x2            x3            x4
## -1.966396e-01  1.203875e-01 -2.161312e+00  4.381297e-06
```

-.1966 <= B1 <= -.0874 .1204 <= B2 <= .4436 -2.1613 <= B3 <= 3.3999 .00000438 <= B4 <= .0000114

With 95% confidence th coefficients for the data will be between the calculated ranges.

c

```
sum_comm$r.squared
```

```
## [1] 0.5847496
```

R^2 shows that approx. 58.5% of the variation in the data is being explained by the model. This isn't a great result.

Problem 6

```
commercial_xh<-read.table("CH06PR20.txt", col.names=c('x1','x2','x3','x4'))
Wbef_2<-qt(1-.05/8,76)
predcom<-predict(commercialfit,newdata=commercial_xh,se.fit = T,level=.95)
varR_2 <- (sum_comm$sigma)^2

Sxx_2_x1 <- sum( commercial_data$x1 * commercial_data$x1) - length(commercial_data$x1) * (mean(commercia
SE_x1 <- sqrt(varR_2*((1/length(commercial_data$x1) + (commercial_xh$x1 - mean(commercial_data$x1))^2/S

Sxx_2_x2 <- sum( commercial_data$x2 * commercial_data$x2) - length(commercial_data$x2) * (mean(commercia
SE_x2 <- sqrt(varR_2*((1/length(commercial_data$x2) + (commercial_xh$x2 - mean(commercial_data$x2))^2/S

Sxx_2_x3 <- sum( commercial_data$x3 * commercial_data$x3) - length(commercial_data$x3) * (mean(commercia
SE_x3 <- sqrt(varR_2*((1/length(commercial_data$x3) + (commercial_xh$x3 - mean(commercial_data$x3))^2/S

Sxx_2_x4 <- sum( (commercial_data$x4) * (commercial_data$x4)) - length(commercial_data$x4) * (mean((comm
```

```
## Warning in (commercial_data$x4) * (commercial_data$x4): NAs produced by integer
## overflow
```

```
SE_x4 <- (varR_2*((1/length(commercial_data$x4) + ((commercial_xh$x4) - mean((commercial_data$x4)))^2/S


predcom$fit[1]-Wbef_2*SE_x1
```

```
## [1] 12.86809 12.87003 12.85604 12.86444
```

```
predcom$fit[1]+Wbef_2*SE_x1
```

```
## [1] 18.72816 18.72622 18.74022 18.73181
```

```
predcom$fit[2]-Wbef_2*SE_x2
```

```
## [1] 13.09527 13.09705 13.09199 13.10001
```

```
predcom$fit[2]+Wbef_2*SE_x2
```

```
## [1] 18.95980 18.95802 18.96308 18.95506
```

```
predcom$fit[3]-Wbef_2*SE_x3
```

```
## [1] 12.96752 12.95198 12.97322 12.96752
```

```
predcom$fit[3]+Wbef_2*SE_x3
```

```
## [1] 18.83393 18.84947 18.82823 18.83393
```

```
predcom$fit[4]-Wbef_2*SE_x4
```

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

```
predcom$fit[4]+Wbef_2*SE_x4
```

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

Problem 7

a

```
sy<-sqrt(sum((commercial_data$y-mean(commercial_data$y))^2)/ (length(commercial_data$y)-1))
y_star<-data.frame(y=(1/(sqrt(length(commercial_data$y)-1)))*((commercial_data$y-mean(commercial_data$y)

s1=sqrt(sum((commercial_data$x1-mean(commercial_data$x1))^2)/ (length(commercial_data$x1)-1))
x1_star<-data.frame(x1=(1/(sqrt(length(commercial_data$x1)-1)))*((commercial_data$x1-mean(commercial_da

s2=sqrt(sum((commercial_data$x2-mean(commercial_data$x2))^2)/ (length(commercial_data$x2)-1))
x2_star<-data.frame(x2=(1/(sqrt(length(commercial_data$x2)-1)))*((commercial_data$x2-mean(commercial_da

s3=sqrt(sum((commercial_data$x3-mean(commercial_data$x3))^2)/ (length(commercial_data$x3)-1))
x3_star<-data.frame(x3=(1/(sqrt(length(commercial_data$x3)-1)))*((commercial_data$x3-mean(commercial_da

s4=sqrt(sum((commercial_data$x4-mean(commercial_data$x4))^2)/ (length(commercial_data$x4)-1))
x4_star<-data.frame(x4=(1/(sqrt(length(commercial_data$x4)-1)))*((commercial_data$x4-mean(commercial_da
transformed_commercial_data<-data.frame(y=y_star,x1=x1_star,x2=x2_star,x3=x3_star,x4=x4_star)
fitted_transformed<-lm(y~., data=transformed_commercial_data)
```

Y*=-.5479x1+.423ex2+.04846x3+.5028x4 standardized regression model

b

The scaled coeffiecient for x2 is .423 and this means this shows that even after scaling the increasing expenses(x2) would increase rates(y)

c

```
b1=(sy/s1)*-.5479
b2=(sy/s2)*.423
b3=(sy/s3)*.04846
b4=(sy/s4)*.5028
b0=mean(commercial_data$y)-b1*(mean(commercial_data$x1))-b2*(mean(commercial_data$x2))-b3*(mean(commerci
b0
```

```
## [1] 12.20475
```

```r
b1
```

```
## [1] -0.1420459
```

```r
b2
```

```
## [1] 0.2815859
```

```r
b3
```

```
## [1] 0.6193261
```

```r
b4
```

```
## [1] 7.924977e-06
```

Yes, the coefficient are the same

Problem 8

a

```r
first_order_brandfit<-lm(y~x1,data=brand_data)
first_order_brandfit
```

```
##
## Call:
## lm(formula = y ~ x1, data = brand_data)
##
## Coefficients:
## (Intercept)           x1
##      50.775        4.425
```

Y=50.775+4.425x1

b

They have the same coefficient

c

```r
anova(first_order_brandfit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## x1         1 1566.45 1566.45  54.751 3.356e-06 ***
## Residuals 14  400.55   28.61
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(first_order_brandfit)
```

```
##
## Call:
## lm(formula = y ~ x1, data = brand_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.475 -4.688 -0.100  4.638  7.525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.775      4.395  11.554 1.52e-08 ***
## x1             4.425      0.598   7.399 3.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.349 on 14 degrees of freedom
## Multiple R-squared:  0.7964, Adjusted R-squared:  0.7818
## F-statistic: 54.75 on 1 and 14 DF,  p-value: 3.356e-06
```

1566.45 for both so yes they are equal.

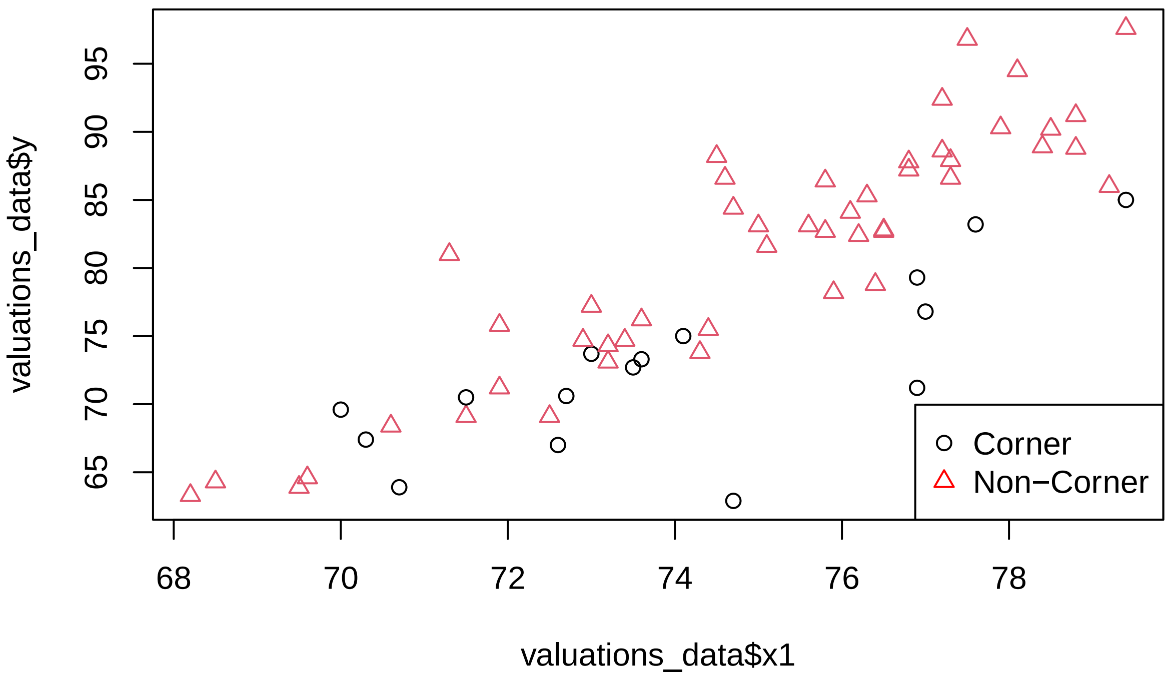
d

The matrix confirms findings in part b and part c because the matrix shows the correlation between x1 and x2 is 0

Problem 9

a

```
valuations_data<-read.table("CH08PR24.txt",col.names = c('y','x1','x2'))
group <- as.factor(ifelse(valuations_data$x2==T, "Group 1", "Group 2"))
plot(valuations_data$x1,valuations_data$y,pch = as.numeric(group), col = group)
legend('bottomright',legend=c('Corner','Non-Corner'), pch=c(1,2), col=c("Black",'Red'))
```



The relation does not appear the same. The non-corner houses look to have a bigger slope.

b

alternatives

H0:B2=B3=0; Ha: not all of the Bk in H0 equal zero.

Decison rule

Partial F test: Reject H0 if F*> F alpha,p-q,n-p (SSR(x2,x3|x1)/(p-q))/MSE(x1,x2,x3)

Conclusion

```
valuationfit<-lm(y~x1+x2+(x1*x2),data=valuations_data)

summary(valuationfit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + (x1 * x2), data = valuations_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8470  -2.1639   0.0913   1.9348   9.9836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -126.9052    14.7225  -8.620 4.33e-12 ***
## x1             2.7759     0.1963  14.142  < 2e-16 ***
## x2            76.0215    30.1314   2.523  0.01430 *
## x1:x2         -1.1075     0.4055  -2.731  0.00828 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.893 on 60 degrees of freedom
## Multiple R-squared:  0.8233, Adjusted R-squared:  0.8145
## F-statistic: 93.21 on 3 and 60 DF,  p-value: < 2.2e-16
```

```
anova(valuationfit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## x1         1 3670.9  3670.9 242.2760 < 2.2e-16 ***
## x2         1  453.1   453.1  29.9073 9.282e-07 ***
## x1:x2      1  113.0   113.0   7.4578  0.008281 **
## Residuals 60  909.1    15.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sum(anova(valuationfit)[2:3,"Sum Sq"])/(4-2)/anova(valuationfit)[4,"Mean Sq"] > qf(1-.05,2,60)
```

```
## [1] TRUE
```

Conclude reject H0.

c

```
pop1_fit<-lm(y~x1,data=valuations_data[valuations_data$x2==T,])
pop2_fit<-lm(y~x1,data = valuations_data[!valuations_data$x2,])

summary(pop1_fit)
```

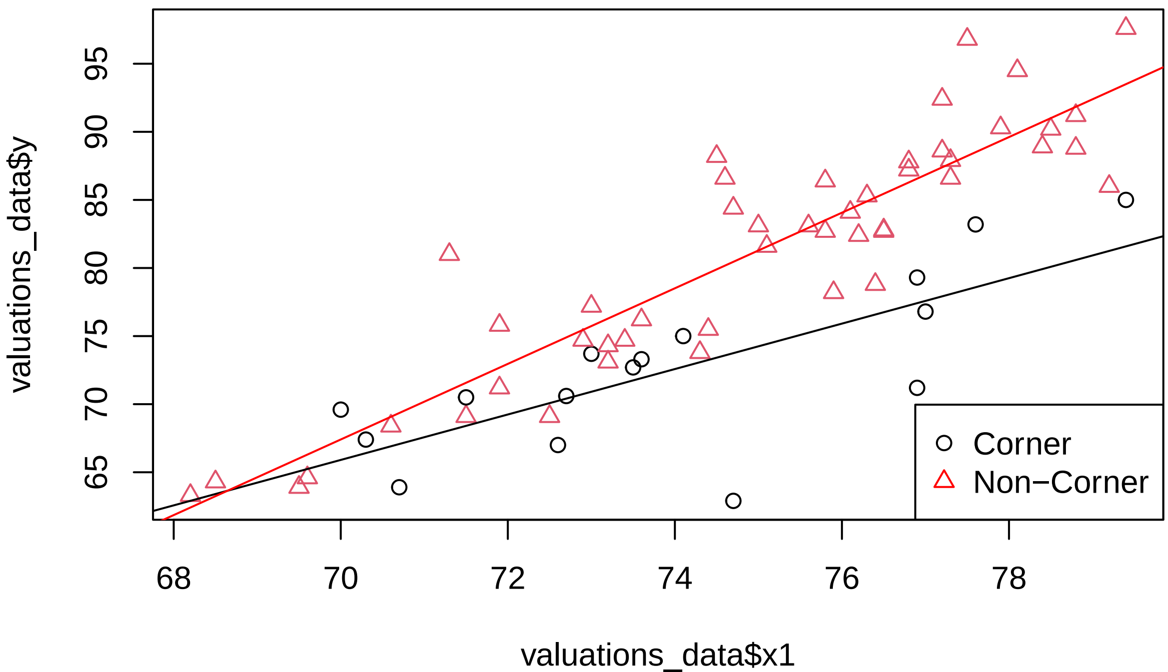
```
##
## Call:
## lm(formula = y ~ x1, data = valuations_data[valuations_data$x2 ==
##     T, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -10.847  -1.382   1.191   2.388   4.615
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -50.8836    28.4687  -1.787 0.095541 .
## x1            1.6684     0.3843   4.342 0.000677 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.215 on 14 degrees of freedom
## Multiple R-squared:  0.5738, Adjusted R-squared:  0.5434
## F-statistic: 18.85 on 1 and 14 DF,  p-value: 0.0006769
```

```
summary(pop2_fit)
```

```
##
## Call:
## lm(formula = y ~ x1, data = valuations_data[!valuations_data$x2,
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9460 -2.1639 -0.6544  1.4775  9.9836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -126.9052    14.3305  -8.856 1.68e-11 ***
## x1             2.7759     0.1911  14.529  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.789 on 46 degrees of freedom
## Multiple R-squared:  0.8211, Adjusted R-squared:  0.8172
## F-statistic: 211.1 on 1 and 46 DF,  p-value: < 2.2e-16
```

```
valuations_data<-read.table("CH08PR24.txt",col.names = c('y','x1','x2'))
group <- as.factor(ifelse(valuations_data$x2==T, "Group 1", "Group 2"))
plot(valuations_data$x1,valuations_data$y,pch = as.numeric(group), col = group)
legend('bottomright',legend=c('Corner','Non-Corner'), pch=c(1,2), col=c("Black",'Red'))
abline(pop1_fit)
abline(pop2_fit,col='red')
```



Y=-50.8836+1.6684x1 (Corner) Y=-126.9052+2.7759x2 (Non-Corner)