Linear Statistical Analysis
Elizaveta Latash
Fall 2018

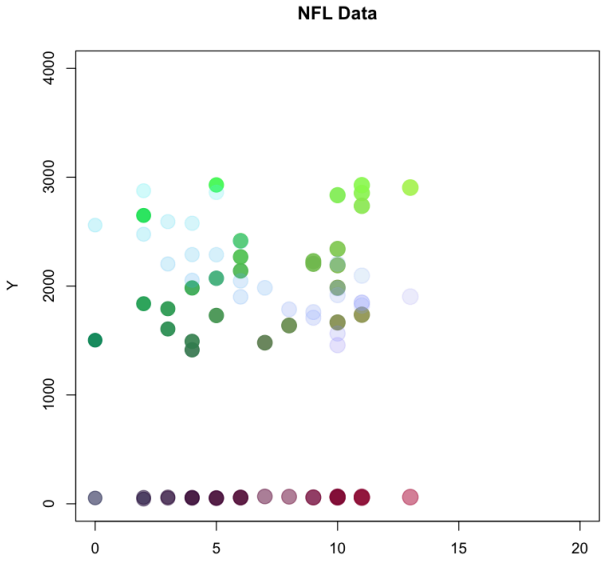
Linear Statistical Analysis : Homework 3

Problem 3.1

Fit a linear regression model relating games won to the team's passing yardage (x2), the percentage of rushing plays (x7), and the opponents yards rushing (x8).

a.

```
NFL=read.csv("data-table-B1.csv")
Games_won=NFL$y
Passing_yardage=NFL$x2
Percent_rush_plays=NFL$x7
Yards_rush=NFL$x8
```




```
a=abs(Games_won)
b=abs(Passing_yardage)
c=abs(Percent_rush_plays)
d=abs(Yards_rush)
plot(a,b , xlim=c(0,20) , ylim=c(0,4000) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/20,b/3000,0.3,0.9) , main="NFL Data",xlab="X",ylab="Y")
points(a,d , xlim=c(0,20) , ylim=c(0,4000) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/20,d/3000,0.99,0.2) , main="Linear Model with N=100, StDev=1",xlab="X",ylab="Y")
> points(a,c , xlim=c(0,20) , ylim=c(0,4000) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/20,c/3000,0.2,0.5) , main="Linear Model with N=100, StDev=1",xlab="X",ylab="Y")
```

```
NFL.Fit=lm(Games_won~ Passing_yardage+ Percent_rush_plays+ Yards_rush)


> NFL.Fit

Call:
lm(formula = Games_won ~ Passing_yardage + Percent_rush_plays +
    Yards_rush, data = NFL)

Coefficients:
       (Intercept)     Passing_yardage  Percent_rush_plays
         -1.808372            0.003598            0.193960
        Yards_rush
         -0.004815
```

~~~~~ $y = -1.808372 + 0.003598\, x_2 + 0.193960\, x_7 - 0.004815\, x_8 x_6$ ~~~~~

b.

```
A.NFL=anova(NFL.Fit)
> A.NFL

Analysis of Variance Table

Response: Games_won
                  Df  Sum Sq Mean Sq F value    Pr(>F)
Passing_yardage    1  76.193  76.193  26.172 3.100e-05 ***
Percent_rush_plays  1 139.501 139.501  47.918 3.698e-07 ***
Yards_rush         1  41.400  41.400  14.221 0.0009378 ***
Residuals         24  69.870   2.911
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1



> NFL.SS=sum(A.NFL[1:3,2])
> NFL.SS
[1] 257.0943
> NFL.MS=NFL.SS/3
> NFL.MS
[1] 85.69809
> NFL.F=NFL.MS/2.911
> NFL.F
[1] 29.4394
> NFL.p=1-pf(NFL.F,3,24)
> NFL.p
[1] 3.270832e-08
> NFL.Total.SS=sum(A.NFL[,2])
> NFL.Total.SS
```
~~~~~

```
[1] 326.9643
```

	df	SS	MS	F	p-value
Reg	3	257.0943	85.69809	29.4394	3.270832e-08
Res	24	69.870	2.911		
Total	27	326.9643			

c.

Calculate t-statistics for Ho: Beta2=0, Ho: Beta7=0, and Ho: Beta8=0.

```
> summary(NFL.Fit)

Call:
lm(formula = Games_won ~ Passing_yardage + Percent_rush_plays +
    Yards_rush)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0370 -0.7129 -0.2043  1.1101  3.7049

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.808372   7.900859  -0.229 0.820899
Passing_yardage     0.003598   0.000695   5.177 2.66e-05 ***
Percent_rush_plays  0.193960   0.088233   2.198 0.037815 *
Yards_rush         -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.706 on 24 degrees of freedom
Multiple R-squared:  0.7863,   Adjusted R-squared:  0.7596
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

Ho: Beta2=0

```
> t_B2=(0.003598-0)/0.000695
> t_B2
[1] 5.176978
> p_B2=2*(1-pt(t_B2,24))
> p_B2
[1] 2.656473e-05
```

Ho: Beta7=0

```
> t_B7=(0.193960-0)/0.088233
> t_B7
[1] 2.19827
> p_B7=2*(1-pt(t_B7,24))
```

```
> p_B7
[1] 0.03781446
```

Ho: Beta8=0

```
> t_B8=(-0.004816-0)/ 0.001277
> t_B8
[1] -3.771339
> p_B8=2*(pt(-abs(t_B8),24))
> p_B8
[1] 0.0009370586
```

d.

Calculate R^2 and R^2 Adjusted

```
R_2=NFL.SS/NFL.Total.SS
> R_2
[1] 0.7863069

R_2_Adj=1-(1-R_2)*(27/24)
> R_2_Adj
[1] 0.7595953
```

e.

Using the partial F test, determine the contribution of x7 to the model. How is this F statistic related to the t-test for B7?

```
NFL.Fit_test_x7=lm(Games_won~Passing_yardage+Yards_rush)

anova(NFL.Fit_test_x7, NFL.Fit)

Analysis of Variance Table

Model 1: Games_won ~ Passing_yardage + Yards_rush
Model 2: Games_won ~ Passing_yardage + Percent_rush_plays + Yards_rush
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     25 83.938
2     24 69.870  1    14.068 4.8324 0.03782 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

With a p-value of .03782 we reject Ho and we favor the full model. This is the same value as the p-valuxe from our comparison of B7 with zero.

Problem 3.4

Reconsider the NFL data. Fit a linear regression model using only x7 and x8 as regressors

a.

```
NFL.Fit.2=lm(Games_won~ Percent_rush_plays+ Yards_rush)

> A.NFL.2=anova(NFL.Fit.2)
> A.NFL.2
Analysis of Variance Table

Response: Games_won
                   Df  Sum Sq Mean Sq F value   Pr(>F)
Percent_rush_plays  1  97.238  97.238  16.437 0.000431 ***
Yards_rush          1  81.828  81.828  13.832 0.001015 **
Residuals          25 147.898   5.916
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1



> NFL.SS.2=sum(A.NFL.2[1:2,2])
> NFL.SS.2
[1] 179.0662
>  NFL.MS.2=NFL.SS.2/2
> NFL.MS.2
[1] 89.53309
> NFL.F.2=NFL.MS.2/5.916
> NFL.F.2
[1] 15.13406
> NFL.p.2=1-pf(NFL.F.2,2,25)
> NFL.p.2
[1] 4.935335e-05
> NFL.Total.SS.2=sum(A.NFL.2[,2])
> NFL.Total.SS.2
[1] 326.9643
```

	df	SS	MS	F	p-value
Reg	2	179.0662	89.53309	15.13406	4.935335e-05
Res	25	147.898	5.916		
Total	27	326.9643			

b.

Calculate R^2 and R^2 Adjusted

```
R_2.2=NFL.SS.2/NFL.Total.SS.2
> R_2.2
[1] 0.5476628

R_2_Adj.2=1-(1-R_2.2)*(27/25)
> R_2_Adj
[1] 0.5114759
```

Including x2 we had

```
> R_2
[1] 0.7863069
> R_2_Adj
[1] 0.7595953
```

These R^2 and adjusted R^2 values show that the inclusion of x2 was important to having a good linear model that explains most of the variation in the data.

c.

Calculate a confidence interval on Beta7 and on the mean number of games won when x7=56 and x8=2100.

```
> summary(NFL.Fit.2)

Call:
lm(formula = Games_won ~ Percent_rush_plays + Yards_rush)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7985 -1.5166 -0.5792  1.9927  4.5248

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        17.944319   9.862484   1.819  0.08084 .
Percent_rush_plays  0.048371   0.119219   0.406  0.68839
Yards_rush         -0.006537   0.001758  -3.719  0.00102 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.432 on 25 degrees of freedom
Multiple R-squared:  0.5477,  Adjusted R-squared:  0.5115
F-statistic: 15.13 on 2 and 25 DF,  p-value: 4.935e-05
```

```
a=qt(0.95,25)
> a
[1] 1.708141
B7hat=0.048371
```

```
B7_sterr=0.119219
> Lower_Bound=B7hat-a*B7_sterr
> Lower_Bound
[1] -0.1552718
> Upper_Bound=B7hat+a*B7_sterr
> Upper_Bound
[1] 0.2520138
```

```
NFL.data.frame.2b=data.frame(Percent_rush_plays=56, Yards_rush=2100)

predict(NFL.Fit.2,NFL.data.frame.2b,interval="confidence")

       fit      lwr      upr

1 6.926243 5.828643 8.023842
```

The confidence interval on Beta7 is (-0.1552718, 0.2520138).

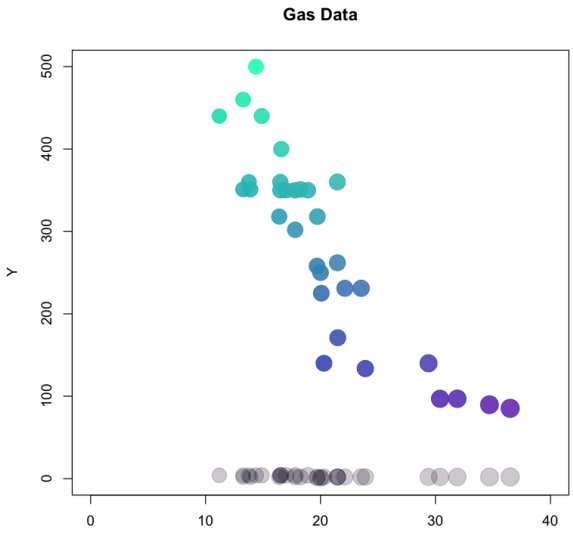
The confidence interval on the mean number of games won when x7=56 and x8=2100 is (5.828643, 8.023842).

Problem 3.5

Fit a linear regression model relating gasoline mileage y (mpg) to engine displacement x1 and the number of carburetor barrels x6.

a.

```
Gas=read.csv("data-table-B3.csv")
MPG=Gas$y
Engine_d=Gas$x1
Carb_barrels=Gas$x6
```




```
a=abs(MPG)
b=abs(Engine_d)
c=abs(Carb_barrels)
plot(a,b , xlim=c(0,40) , ylim=c(0,500) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/90,b/500,0.7,0.9) , main="Gas Data",xlab="X",ylab="Y")
points(a,c , xlim=c(0,40) , ylim=c(0,500) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/400,c/500,0.1,0.2) , main="Gas Data",xlab="X",ylab="Y")
```

```
Gas.fit=lm(MPG~Engine_d+Carb_barrels)
```

```
Call:
lm(formula = MPG ~ Engine_d + Carb_barrels)

Coefficients:
 (Intercept)      Engine_d  Carb_barrels
    32.88455      -0.05315       0.95922
```

~~~~~ $y = 32.88455 - 0.05315\, x_1 + 0.95922\, x_6$ ~~~~~

b.

```
> A.Gas=anova(Gas.fit)
> A.Gas
Analysis of Variance Table

Response: MPG
             Df Sum Sq Mean Sq F value    Pr(>F)
Engine_d      1 955.72  955.72 105.290 3.666e-11 ***
Carb_barrels  1  18.59   18.59   2.048    0.1631
```
~~~~~

```
Residuals    29 263.23    9.08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> Gas.SS=sum(A.Gas[1:2,2])
> Gas.SS
[1] 974.3095
> Gas.MS=Gas.SS/2
> Gas.MS
[1] 487.1548
> Gas.F=Gas.MS/9.08
> Gas.F
[1] 53.65141
> Gas.p=1-pf(Gas.F,2,29)
> Gas.p
[1] 1.796598e-10
> Gas.Total.SS=sum(A.Gas[,2])
> Gas.Total.SS
[1] 1237.544
```

	df	SS	MS	F	p-value
Reg	2	974.3095	487.1548	53.65141	1.796598e-10
Res	29	263.23	9.08		
Total	31	1237.544			

c.

Calculate R^2 and R^2 Adjusted

```
R_2=Gas.SS/Gas.Total.SS
> R_2
[1] 0.7872928

R_2_Adj=1-(1-R_2)*(31/29)
> R_2_Adj
[1] 0.7726233
```

This is an improvement, when compared to the model from Problem 2.4.

d.

Calculate the CI on Beta1.

```
> summary(Gas.fit)

Call:
lm(formula = MPG ~ Engine_d + Carb_barrels)

Residuals:
    Min      1Q  Median      3Q     Max
-7.0623 -1.6687 -0.3628  1.6221  6.2305

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.884551   1.535408  21.417  < 2e-16 ***
Engine_d     -0.053148   0.006137  -8.660 1.55e-09 ***
Carb_barrels  0.959223   0.670277   1.431    0.163

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

Residual standard error: 3.013 on 29 degrees of freedom
Multiple R-squared:  0.7873,   Adjusted R-squared:  0.7726
F-statistic: 53.67 on 2 and 29 DF,  p-value: 1.79e-10
```

```
a=qt(0.95,29)
> a
[1] 1.699127
B1hat=-0.053148
B1_sterr=0.006137
> Lower_Bound=B1hat-a*B1_sterr
> Lower_Bound
[1] -0.0635755
> Upper_Bound=B1hat+a*B1_sterr
> Upper_Bound
[1] -0.04272046
```

The confidence interval on Beta1 is (-0.0635755, -0.04272046).

e.

Calculate t-statistics for Ho: Beta1=0 and Ho: Beta6=0.

Ho: Beta1=0

```
> t_B1=(-0.053148-0)/ 0.006137
> t_B1
[1] -8.660257
> p_B1=2*(pt(-abs(t_B1),29))
> p_B1
[1] 1.550599e-09
```

Ho: Beta6=0

```
> t_B6=(0.959223-0)/ 0.670277
> t_B6
[1] 1.431084
> p_B6=2*(1-pt(t_B6,29))
> p_B6
[1] 0.1630948
```

f.

Calculate a confidence interval on mean gas mileage when x1=275in^3 and x6=2 barrels.

```
Gas.data.frame.2=data.frame(Engine_d=275, Carb_barrels=2)

predict(Gas.fit,Gas.data.frame.2,interval="confidence")

       fit      lwr      upr
1 20.18739 18.87221 21.50257
```

The confidence interval on mean gas mileage when x1=275in^3 and x6=2 barrels is (18.87221, 21.50257).

g.

Calculate a prediction interval on mean gas mileage when x1=275in^3 and x6=2 barrels.

```
Gas.data.frame.2=data.frame(Engine_d=275, Carb_barrels=2)

predict(Gas.fit,Gas.data.frame.2,interval="prediction")

       fit     lwr      upr
1 20.18739 13.8867 26.48808
```

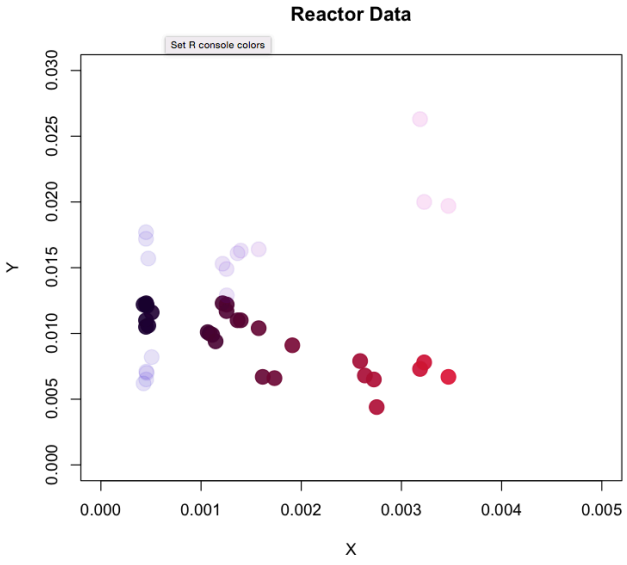
The confidence interval on mean gas mileage when x1=275in^3 and x6=2 barrels is (13.8867, 26.48808).

Problem 3.9

a.

Fit a linear regression model relating NbOCl3(y) to concentration of COCL2(x1) and mole fraction (x4).

```
Reactor=read.csv("data-table-B6.csv")
NbOCl3=Reactor$y
COCL2=Reactor$x1
Mole_fraction=Reactor$x4
```




```
a=abs(NbOCl3)
b=abs(COCL2)
c=abs(Mole_fraction)
plot(a,b , xlim=c(0,.005) , ylim=c(0,.03) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/.004,b/500,0.2,0.9) , main="Reactor Data",xlab="X",ylab="Y")
points(a,c , xlim=c(0,.005) , ylim=c(0,.03) , pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/.004,b/900,0.8,0.1) , main="Reactor Data",xlab="X",ylab="Y")
```

```
Reactor.fit=lm(NbOCl3~COCL2+Mole_fraction)
> Reactor.fit

Call:
lm(formula = NbOCl3 ~ COCL2 + Mole_fraction)

Coefficients:
  (Intercept)          COCL2  Mole_fraction
     0.004833      -0.344984       -0.000143
```

~~~~~ $y = 0.004833 - 0.344984\,x_1 - 0.000143\,x_4$ ~~~~~

b.

Test for significance of regression

```
> A.Reactor=anova(Reactor.fit)
> A.Reactor
Analysis of Variance Table

Response: NbOCl3
              Df     Sum Sq    Mean Sq F value   Pr(>F)
COCL2          1 1.6615e-05 1.6615e-05 49.3177 2.32e-07 ***
Mole_fraction  1 1.0000e-10 1.0000e-10  0.0003   0.9855
Residuals     25 8.4222e-06 3.3690e-07
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


> Reactor.SS=sum(A.Reactor [1:2,2])
> Reactor.SS
[1] 1.661466e-05
> Reactor.MS= Reactor.SS/2
> Reactor.MS
[1] 8.307331e-06
> Reactor.F= Reactor.MS/3.3690e-07
> Reactor.F
[1] 24.65815
> Reactor.p=1-pf(Reactor.F,2,25)
> Reactor.p
[1] 1.218145e-06
> Reactor.Total.SS=sum(A.Reactor[,2])
> Reactor.Total.SS
[1] 2.503686e-05
```

	df	SS	MS	F	p-value
Reg	2	1.661466e-05	8.307331e-06	24.65815	1.218145e-06
Res	25	8.4222e-06	3.3690e-07		
Total	27	2.503686e-05			

c.

Calculate R^2 and R^2 Adjusted
~~~~~

```
R_2=Reactor.SS/ Reactor.Total.SS
> R_2
[1] 0.663608

R_2_Adj=1-(1-R_2)*(27/25)
> R_2_Adj
[1] 0.6366966
```

d.

Using t-tests, determine the contribution of x1 and x4 to the model. Are both necessary? We test this by calculating t-statistics for Ho: Beta1=0 and Ho: Beta4=0.

```
> summary(Reactor.fit)

Call:
lm(formula = NbOCl3 ~ COCL2 + Mole_fraction)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0009015 -0.0003526 -0.0001538  0.0003847  0.0010874

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0048333  0.0008142   5.936 3.39e-06 ***
COCL2         -0.3449837  0.0673963  -5.119 2.74e-05 ***
Mole_fraction -0.0001430  0.0078151  -0.018    0.986
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0005804 on 25 degrees of freedom
Multiple R-squared:  0.6636,   Adjusted R-squared:  0.6367
F-statistic: 24.66 on 2 and 25 DF,  p-value: 1.218e-06
```

Ho: Beta1=0

```
> t_B1=(-0.3449837-0)/ 0.0673963
> t_B1
[1] -5.118734
> p_B1=2*(pt(-abs(t_B1),25))
> p_B1
[1] 2.742025e-05
```

Ho: Beta4=0

```
> t_B4=(-0.0001430-0)/0.0078151
```

```
> t_B4
[1] -0.01829791
> p_B4=2*(pt(-abs(t_B4),25))
> p_B4
[1] 0.9855464
```

We conclude that B1 is significantly different from zero, while B4 is not. Maybe the model will work well with only B1.

e. Multicollinearity

```
> install.packages("VIF")
Warning: unable to access index for repository
https://mirrors.nics.utk.edu/cran/src/contrib
Warning: unable to access index for repository
https://mirrors.nics.utk.edu/cran/bin/macosx/contrib/3.2
Warning message:
package 'VIF' is not available (for R version 3.2.1)
```

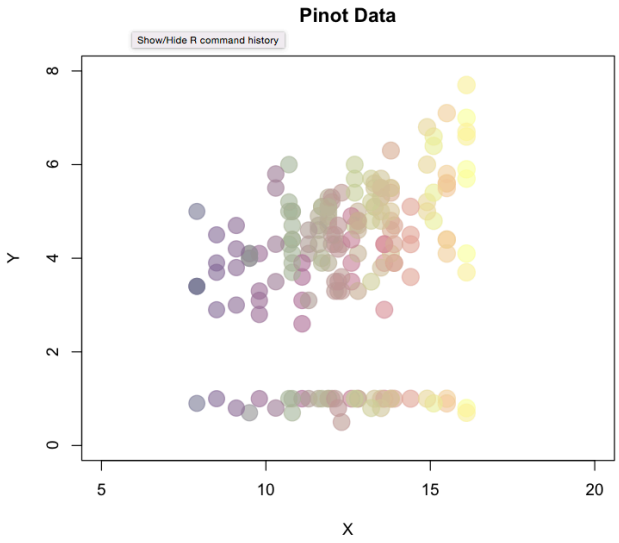
Tried changing mirrors and that installed it as a binary. Neither the VIF, nor the car package gave me the vif function

Problem 3.10

a.

Fit a linear regression model pinot noir quality to clarity, aroma ,body ,flavor ,and oakiness.

```
Pinot=read.csv("data-table-B11.csv")
Quality=Pinot$Quality
Clarity=Pinot$Clarity
Aroma=Pinot$Aroma
Body=Pinot$Body
Flavor=Pinot$Flavor
Oakiness=Pinot$Oakiness
```




```
a=abs(Quality)
b=abs(Flavor)
c=abs(Oakiness)
d=abs(Clarity)
e=abs(Body)
f=abs(Aroma)
plot(a,b , xlim=c(5,20) , ylim=c(0,8), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/16.2,b/7,0.6,0.6) , main="Pinot Data",xlab="X",ylab="Y")
points(a,c , xlim=c(5,20) , ylim=c(0,8), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/16.2,b/7,0.6,0.6) , main="Pinot Data",xlab="X",ylab="Y")
points(a,d , xlim=c(5,20) , ylim=c(0,8), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/16.2,b/7,0.6,0.6) , main="Pinot Data",xlab="X",ylab="Y")
points(a,e , xlim=c(5,20) , ylim=c(0,8), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/16.2,b/7,0.6,0.6) , main="Pinot Data",xlab="X",ylab="Y")
points(a,f , xlim=c(5,20) , ylim=c(0,8), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/16.2,b/7,0.6,0.6) , main="Pinot Data",xlab="X",ylab="Y")
```

```
Pinot.fit=lm(Quality~ Clarity + Aroma + Body + Flavor + Oakiness)
```

```
> Pinot.fit

Call:
lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness)

Coefficients:
(Intercept)      Clarity        Aroma         Body       Flavor
     3.9969       2.3395       0.4826       0.2732       1.1683
   Oakiness
    -0.6840
```

~~~~~ $y = 3.9969 + 2.3395\,x_1 + 0.4826\,x_2 + 0.2732\,x_3 + 1.1683\,x_4 - 0.6840\,x_5$ ~~~~~
~~~~~

b.

Test for significance of regression

```
> A.Pinot=anova(Pinot.fit)
> A.Pinot
Analysis of Variance Table

Response: Quality
          Df Sum Sq Mean Sq F value    Pr(>F)
Clarity    1  0.125   0.125  0.0926 0.7628120
Aroma      1 77.353  77.353 57.2351 1.286e-08 ***
Body       1  6.414   6.414  4.7461 0.0368417 *
Flavor     1 19.050  19.050 14.0953 0.0006946 ***
Oakiness   1  8.598   8.598  6.3616 0.0168327 *
Residuals 32 43.248   1.352
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> Pinot.SS=sum(A.Pinot[1:5,2])
> Pinot.SS
[1] 111.5404
> Pinot.MS= Pinot.SS/5
> Pinot.MS
[1] 22.30808
> Pinot.F= Pinot.MS/1.352
> Pinot.F
[1] 16.50006
> Pinot.p=1-pf(Pinot.F,5,32)
> Pinot.p
[1] 4.722458e-08
> Pinot.Total.SS=sum(A.Pinot[,2])
> Pinot.Total.SS
[1] 154.7884
```

	df	SS	MS	F	p-value
Reg	5	111.5404	22.30808	16.50006	4.722458e-08
Res	32	43.248	1.352		
Total	37	154.7884			

We find that the model is significant, given the very small p-value.

c.

Using t-tests, determine the contribution of each regressor to the model. We test this by calculating t-statistics for Ho: Beta1=0, Ho: Beta2=0, Ho: Beta3=0, Ho: Beta4=0, and Ho: Beta5=0.

```
> summary(Pinot.fit)

Call:
lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness)

Residuals:
     Min       1Q   Median       3Q      Max
-2.85552 -0.57448 -0.07092  0.67275  1.68093

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.9969     2.2318   1.791 0.082775 .
Clarity       2.3395     1.7348   1.349 0.186958
Aroma         0.4826     0.2724   1.771 0.086058 .
Body          0.2732     0.3326   0.821 0.417503
Flavor        1.1683     0.3045   3.837 0.000552 ***
Oakiness     -0.6840     0.2712  -2.522 0.016833 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.163 on 32 degrees of freedom
Multiple R-squared:  0.7206,   Adjusted R-squared:  0.6769
F-statistic: 16.51 on 5 and 32 DF,  p-value: 4.703e-08
```

Ho: Beta1=0

```
Clarity       2.3395     1.7348
> t_B1=(2.3395-0)/1.7348
> t_B1
[1] 1.34857
> p_B1=2*(pt(-abs(t_B1),32))
> p_B1
[1] 0.1869426
```

Ho: Beta2=0

```
Aroma         0.4826     0.2724
> t_B2=(0.4826-0)/ 0.2724
> t_B2
[1] 1.771659
> p_B2=2*(pt(-abs(t_B2),32))
> p_B2
[1] 0.08597482
```

Ho: Beta3=0

```
Body          0.2732     0.3326
> t_B3=(0.2732-0)/ 0.3326
> t_B3
```

```
[1] 0.8214071
> p_B3=2*(pt(-abs(t_B3),32))
> p_B3
[1] 0.4174924
```

Ho: Beta4=0

```
Flavor        1.1683     0.3045
> t_B4=(1.1683-0)/ 0.3045
> t_B4
[1] 3.836782
> p_B4=2*(pt(-abs(t_B4),32))
> p_B4
[1] 0.0005527262
```

Ho: Beta5=0

```
Oakiness     -0.6840     0.2712
> t_B5=(-0.6840-0)/ 0.2712
> t_B5
[1] -2.522124
> p_B5=2*(pt(-abs(t_B5),32))
> p_B5
[1] 0.01683689
```

We conclude that B5, B4 and maybe B2 are different from zero, while B1 and B3 are not different from zero. Removing B1 and B3 and retesting might be a good idea. B2 and B4 are the most important to keep.

d.

Calculate R^2 and R^2 Adjusted for the full and reduced model with only B2 and B4 (aroma and flavor)

```
R_2.full=Pinot.SS/ Pinot.Total.SS
> R_2.full
[1] 0.7205992

R_2_Adj.full=1-(1-R_2.full)*(37/32)
> R_2_Adj.full
[1] 0.6769428


Pinot.fit.reduced= lm(Quality~ Aroma + Flavor)
A.Pinot.reduced=anova(Pinot.fit.reduced)
Pinot.reduced.Total.SS=sum(A.Pinot.reduced[,2])
Pinot.reduced.SS= sum(A.Pinot.reduced[1:2,2])
```

```
R_2.reduced=Pinot.reduced.SS / Pinot.reduced.Total.SS
> R_2.reduced
[1] 0.6585515

R_2_Adj.reduced =1-(1-R_2.reduced)*(37/35)
> R_2_Adj.reduced
[1] 0.6390402
```

The R^2 and R^2 Adj went down in the reduced model. That means that one of the other regressors help predict variability in the data. I would add B5:Oakiness back into the model because B5 was found to be significantly different from zero.

e.

Calculate the CI on flavor for both the full and reduced model. In the full model flavor is B4 and in the reduced model flavor is B2.

```
> summary(Pinot.fit.reduced)

Call:
lm(formula = Quality ~ Aroma + Flavor)

Residuals:
     Min       1Q   Median       3Q      Max
-2.19048 -0.60300 -0.03203  0.66039  2.46287

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.3462     1.0091   4.307 0.000127 ***
Aroma         0.5180     0.2759   1.877 0.068849 .
Flavor        1.1702     0.2905   4.027 0.000288 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.229 on 35 degrees of freedom
Multiple R-squared:  0.6586,  Adjusted R-squared:  0.639
F-statistic: 33.75 on 2 and 35 DF,  p-value: 6.811e-09
```

```
Full model :    Flavor        1.1683     0.3045

> a.full=qt(0.95,32)
> a.full
[1] 1.693889
```

```
> B4hat=1.1683
> B4_sterr=0.3045
> Lower_Bound=B4hat-a.full*B4_sterr
> Lower_Bound
[1] 0.6525109
> Upper_Bound=B4hat+a.full*B4_sterr
> Upper_Bound
[1] 1.684089
```

The confidence interval on Beta4 in the full model is (0.6525109, 1.684089).

Full model : Flavor 1.1702 0.2905

```
> a.reduced =qt(0.95,35)
> a.reduced
[1] 1.693889
> B2hat=1.1702
> B2_sterr=0.2905
> Lower_Bound=B2hat-a.reduced*B2_sterr
> Lower_Bound
[1] 0.6793792
> Upper_Bound=B2hat+a.reduced*B2_sterr
> Upper_Bound
[1] 1.661021
```

The confidence interval on Beta2 in the reduced model is (0.6793792, 1.661021).

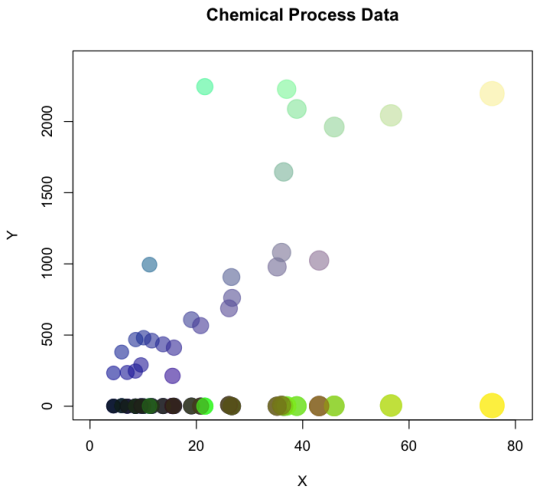
The interval is slightly larger in the full model.

Problem 3.8

a.

Fit a linear regression model relating CO2 product(y) to the total solvent (x6) and hydrogen consumption (x7).

```
Chem_process=read.csv("data-table-B5.csv")
CO2=Chem_process$y
Solvent=Chem_process$x6
Hydrogen=Chem_process$x7
```




```
a=abs(CO2)
b=abs(Solvent)
c=abs(Hydrogen)
```

```
plot(a,b , xlim=c(0,80) , ylim=c(0,2400), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/76,b/2300,0.6,0.6) , main="Chemical Process Data",xlab="X",ylab="Y")
points(a,c , xlim=c(0,80) , ylim=c(0,2400), pch=20 , bg="white" , cex=3+(a/30) ,
col=rgb(a/76,b/2300,0.1,0.6) , main="Chemical Process Data",xlab="X",ylab="Y")
```

```
> Chem_Process.fit=lm(CO2~Solvent+Hydrogen)
> Chem_Process.fit

Call:
lm(formula = CO2 ~ Solvent + Hydrogen)

Coefficients:
(Intercept)      Solvent     Hydrogen
    2.52646      0.01852      2.18575
```

~~~~~ $y = 2.52646 + 0.01852\ x_6 + 2.18575\ x_7$ ~~~~~

b.

Test for significance of regression

```
> A.Chem_Process=anova(Chem_Process.fit)
> A.Chem_Process
Analysis of Variance Table

Response: CO2
          Df Sum Sq Mean Sq F value    Pr(>F)
Solvent    1 5008.9  5008.9 50.8557 2.267e-07 ***
Hydrogen   1  497.3   497.3  5.0495    0.0341 *
```
~~~~~

```
Residuals 24 2363.8    98.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


> Chem_Process.SS=sum(A.Chem_Process[1:2,2])
> Chem_Process.SS
[1] 5506.277
> Chem_Process.MS= Chem_Process.SS/2
> Chem_Process.MS
[1] 2753.138
> Chem_Process.F= Chem_Process.MS/98.5
> Chem_Process.F
[1] 27.95064
> Chem_Process.p=1-pf(Chem_Process.F,2,24)
> Chem_Process.p
[1] 5.393734e-07
> Chem_Process.Total.SS=sum(A.Chem_Process[,2])
> Chem_Process.Total.SS
[1] 7870.112
```

	df	SS	MS	F	p-value
Reg	2	5506.277	2753.138	27.95064	5.393734e-07
Res	24	2363.8	98.5		
Total	26	7870.112			

We find that the model is significant, given the very small p-value.

```
R_2=Chem_Process.SS/Chem_Process.Total.SS
> R_2
[1] 0.699644

R_2_Adj=1-(1-R_2)*(26/24)
> R_2_Adj
[1] 0.6746144
```

c.

Using t-tests, determine the contribution of each regressor to the model. We test this by calculating t-statistics for Ho: Beta6=0 and Ho: Beta7=0.

```
> summary(Chem_Process.fit)

Call:
lm(formula = CO2 ~ Solvent + Hydrogen)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-23.2035  -4.3713   0.2513   4.9339  21.9682

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.526460   3.610055   0.700   0.4908
Solvent     0.018522   0.002747   6.742 5.66e-07 ***
Hydrogen    2.185753   0.972696   2.247   0.0341 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.924 on 24 degrees of freedom
Multiple R-squared:  0.6996,   Adjusted R-squared:  0.6746
F-statistic: 27.95 on 2 and 24 DF,  p-value: 5.391e-07
```

Ho: Beta6=0

```
Solvent     0.018522   0.002747
> t_B6=(0.018522-0)/ 0.002747
> t_B6
[1] 6.742628
> p_B6=2*(pt(-abs(t_B6),24))
> p_B6
[1] 5.654695e-07
```

Ho: Beta7=0

```
Hydrogen    2.185753   0.972696
> t_B7=(2.185753-0)/ 0.972696
> t_B7
[1] 2.247108
> p_B7=2*(pt(-abs(t_B7),24))
> p_B7
[1] 0.03409783
```

We conclude that both B6 and B7 contribute to the model.

d.

Construct a 95% CI on B6 and B7.

```
Solvent     0.018522   0.002747
a=qt(0.95,24)
> a
[1] 1.710882
B6hat= 0.018522
B6_sterr=0.002747
> Lower_Bound=B6hat-a*B6_sterr
> Lower_Bound
[1] 0.01382221
> Upper_Bound=B6hat+a*B6_sterr
> Upper_Bound
[1] 0.02322179
```

```
Hydrogen    2.185753   0.972696
a=qt(.95,24)
> a
[1] 1.710882
B7hat=2.185753
B7_sterr=0.972696
> Lower_Bound=B7hat-a*B7_sterr
> Lower_Bound
[1] 0.5215848
> Upper_Bound=B7hat+a*B7_sterr
> Upper_Bound
[1] 3.849921
```

The confidence interval on B6 is (0.01382221, 0.02322179).

The confidence interval on B7 is (0.5215848, 3.849921).

e.

Refit the model using only x6 as the regressor. Test for significance of regression and calculate R^2 and RAdj^2.

```
Chem_Process.fit.reduced=lm(CO2~Solvent)


A.Chem_process.reduced=anova(Chem_Process.fit.reduced)

Analysis of Variance Table

Response: CO2
          Df Sum Sq Mean Sq F value    Pr(>F)
Solvent    1 5008.9  5008.9  43.766 6.238e-07 ***
Residuals 25 2861.2   114.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1



Chem.reduced.SS=sum(A.Chem_process.reduced[1,2])
>  Chem.reduced.SS
[1] 5008.936
Chem.reduced.MS=Chem.reduced.SS
Chem.reduced.F=Chem.reduced.MS/114.4
> Chem.reduced.F
[1] 43.78441
Chem.reduced.p=1-pf(Chem.reduced.F,1,25)
> Chem.reduced.p
[1] 6.216762e-07
Chem.reduced.Total.SS=sum(A.Chem_process.reduced[,2])
```

```
> Chem.reduced.Total.SS
[1] 7870.112
```

```
> Chem_Process.p=1-pf(Chem_Process.F,2,24)
> Chem_Process.p
[1] 5.393734e-07
```

	df	SS	MS	F	p-value
Reg	1	5008.9	2753.138	43.78441	6.216762e-07
Res	25	2861.2	114.4		
Total	26	7870.112			

Our model is still significant, but the p-value is larger compared to the full model so it seems the B7 regressor made full the model better.

```
R_2.reduced= Chem.reduced.SS /Chem.reduced.Total.SS
> R_2.reduced
[1] 0.6364504

R_2_Adj.reduced =1-(1-R_2.reduced)*(26/25)
> R_2_Adj.reduced
[1] 0.6219084
```

The R values are lower than in the full model, and so the x6 regressor does not explain as much variability in the data as do x6 and x7.

Problem 3.11

a.

Fit a linear regression model relating yield of oil per batch of peanuts(y) to CO2 pressure, CO2 temperature, peanut moisture, CO2 flow rate, and peanut particle size.

```
Peanut=read.csv("data-table-B7.csv")
Oil=Peanut$y
CO2_pressure=Peanut$x1
CO2_temperature=Peanut$x2
Peanut_moisture=Peanut$x3
CO2_flow_rate=Peanut$x4
Particle_size=Peanut$x5
```

```
Peanut.fit=lm(Oil~ CO2_pressure + CO2_temperature + Peanut_moisture +
CO2_flow_rate + Particle_size)
```

```
> Peanut.fit

Call:
lm(formula = Oil ~ CO2_pressure + CO2_temperature + Peanut_moisture +
    CO2_flow_rate + Particle_size)

Coefficients:
    (Intercept)     CO2_pressure  CO2_temperature  Peanut_moisture
      5.208e+01        5.556e-02        2.821e-01        1.250e-01
  CO2_flow_rate    Particle_size
      1.776e-16       -1.606e+01
```

$$y = 5.208e+01 + 5.556e-02\,\boldsymbol{x_1} + 2.821e-01\,\boldsymbol{x_2} + 1.250e-01\,\boldsymbol{x_3} + 1.776e-16\,\boldsymbol{x_4} + -1.606e+01\,\boldsymbol{x_5}$$

b.

Test for significance of regression

```
> A.Peanut=anova(Peanut.fit)

> A.Peanut
Analysis of Variance Table

Response: Oil
                Df Sum Sq Mean Sq  F value    Pr(>F)
CO2_pressure     1  225.0   225.0   3.4589 0.0925445 .
CO2_temperature  1 1560.2  1560.2  23.9854 0.0006254 ***
Peanut_moisture  1    6.2     6.2   0.0961 0.7629488
CO2_flow_rate    1    0.0     0.0   0.0000 1.0000000
Particle_size    1 7921.0  7921.0 121.7679 6.401e-07 ***
Residuals       10  650.5    65.1
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Peanut.SS=sum(A.Peanut[1:5,2])
> Peanut.SS
[1] 9712.5
Peanut.MS=Peanut.SS/5
> Peanut.MS
[1] 1942.5
Peanut.F=Peanut.MS/65.1
> Peanut.F
[1] 29.83871
Peanut.p=1-pf(Peanut.F,5,10)
> Peanut.p
[1] 1.058464e-05
Peanut.Total.SS=sum(A.Peanut[,2])
> Peanut.Total.SS
[1] 10363
```

	df	SS	MS	F	p-value
Reg	5	9712.5	1942.5	29.83871	1.058464e-05
Res	10	650.5	65.1		
Total	15	10363			

We find that the model is significant, given the very small p-value.

c.

Using t-tests, determine the contribution of each regressor to the model. We test this by calculating t-statistics for B1:B5.

```
> summary(Peanut.fit)

Call:
lm(formula = Oil ~ CO2_pressure + CO2_temperature + Peanut_moisture +
    CO2_flow_rate + Particle_size)

Residuals:
    Min      1Q  Median      3Q     Max
-12.250  -4.438   0.125   5.250   9.500

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      5.208e+01  1.889e+01   2.757 0.020218 *
CO2_pressure     5.556e-02  2.987e-02   1.860 0.092544 .
CO2_temperature  2.821e-01  5.761e-02   4.897 0.000625 ***
```

```
Peanut_moisture  1.250e-01  4.033e-01   0.310 0.762949
CO2_flow_rate    1.776e-16  2.016e-01   0.000 1.000000
Particle_size   -1.606e+01  1.456e+00 -11.035  6.4e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.065 on 10 degrees of freedom
Multiple R-squared:  0.9372,   Adjusted R-squared:  0.9058
F-statistic: 29.86 on 5 and 10 DF,  p-value: 1.055e-05
```

Ho: Beta1=0

```
CO2_pressure     5.556e-02  2.987e-02
t_B1=(5.556e-02-0)/2.987e-02
> t_B1
[1] 1.86006
p_B1=2*(pt(-abs(t_B1),10))
> p_B1
[1] 0.09250584
```

Ho: Beta2=0

```
CO2_temperature  2.821e-01  5.761e-02
t_B2=(2.821e-01-0)/ 5.761e-02
> t_B2
[1] 4.896719
p_B2=2*(pt(-abs(t_B2),10))
> p_B2
[1] 0.0006261662
```

Ho: Beta3=0

```
Peanut_moisture  1.250e-01  4.033e-01
t_B3=(1.250e-01-0)/ 4.033e-01
> t_B3
[1] 0.309943
p_B3=2*(pt(-abs(t_B3),10))
> p_B3
[1] 0.762967
```

Ho: Beta4=0

```
CO2_flow_rate    1.776e-16  2.016e-01
t_B4=(1.776e-16-0)/ 2.016e-01
> t_B4
[1] 8.809524e-16
p_B4=2*(pt(-abs(t_B4),10))
> p_B4
[1] 1
```

Ho: Beta5=0

```
Particle_size   -1.606e+01  1.456e+00
t_B5=(-1.606e+01-0)/ 1.456e+00
> t_B5
[1] -11.03022
p_B5=2*(pt(-abs(t_B5),10))
> p_B5
[1] 6.426216e-07
```

We conclude that B2 and B5 contribute to the model. B1 probably contributes to the model, while B3 and B4 do not.

d.

Calculate R^2 and RAdj^2 and compare these to the R^2 and RAdj^2 for a reduced model comparing yield to temperature (x2) and particle size (x5).

```
R_2=Peanut.SS/Peanut.Total.SS
R_2
[1] 0.9372286

R_2_Adj=1-(1-R_2)*(10/15)
R_2_Adj
[1] 0.9581524
```

```
Peanut.fit.reduced=lm(Oil~ CO2_temperature + Particle_size)
A.Peanut.reduced=anova(Peanut.fit.reduced)
Peanut.SS.reduced=sum(A.Peanut.reduced[1:2,2])
Peanut.Total.SS.reduced= sum(A.Peanut.reduced[,2])
```

```
summary(Peanut.fit.reduced)
```

```
Call:
lm(formula = Oil ~ CO2_temperature + Particle_size)

Residuals:
    Min      1Q  Median      3Q     Max
-15.375  -4.188  -0.875   3.438  12.625

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      80.13461    5.69146  14.080 3.01e-09 ***
CO2_temperature   0.28214    0.05883   4.796 0.000349 ***
Particle_size   -16.06498    1.48659 -10.807 7.26e-08 ***
```

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

Residual standard error: 8.236 on 13 degrees of freedom
Multiple R-squared:  0.9149,   Adjusted R-squared:  0.9018
F-statistic: 69.89 on 2 and 13 DF,  p-value: 1.107e-07



R_2.reduced=Peanut.SS.reduced /Peanut.Total.SS.reduced
R_2.reduced
[1] 0.9149136

R_2_Adj.reduced =1-(1-R_2.reduced)*(13/15)
R_2_Adj.reduced
[1] 0.9262585  …. Or 0.9018  by the summary of the fit above
```

There is very little difference between the R^2 values in the full and reduced model. So, we could use the reduced model instead.

e.

Construct a 95% CI on temperature (x2) for the full and reduced models.

Full model:

```
CO2_temperature  2.821e-01  5.761e-02
a=qt(.95,10)
> a
[1] 1.812461
B2hat=2.821e-01
B2_sterr=5.761e-02
Lower_Bound=B2hat-a*B2_sterr
Lower_Bound
[1] 0.1776841
Upper_Bound=B2hat+a*B2_sterr
Upper_Bound
[1] 0.3865159
```

Reduced model:

```
CO2_temperature   0.28214    0.05883
a=qt(.95,10)
> a
[1] 1.812461
B2hat=0.28214
B2_sterr=0.05883
Lower_Bound=B2hat-a*B2_sterr
Lower_Bound
[1] 0.1755129
```

```
Upper_Bound=B2hat+a*B2_sterr
Upper_Bound
[1] 0.3887671
```

There is not a big difference in the confidence interval in the full and reduced model.

Problem 3.12

a.

Fit a linear regression model relating clathrates to amount of surfactant and time.

```
Clathrates=read.csv("data-table-B8.csv")
Clath=Clathrates$y
Surfactant= Clathrates $x1
Time= Clathrates $x2


Clathrates.fit=lm(Clath~ Surfactant + Time)

Clathrates.fit


Call:
lm(formula = Clath ~ Surfactant + Time)

Coefficients:
(Intercept)   Surfactant         Time
    11.0870     350.1192       0.1089
```

~~~~~~ $y = 11.0870 + 350.1192\,x_1 + 0.1089\,x_2$ ~~~~~~

b.

Test for significance of regression

```
A.Clathrates=anova(Clathrates.fit)
A.Clathrates
```
~~~~~~

```
Analysis of Variance Table

Response: Clath
           Df  Sum Sq Mean Sq F value    Pr(>F)
Surfactant  1 1283.90 1283.90  56.137 1.295e-08 ***
Time        1 2723.17 2723.17 119.066 1.742e-12 ***
Residuals  33  754.74   22.87
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Clath.SS=sum(A.Clathrates[1:2,2])
Clath.SS
[1] 4007.072
Clath.MS=Clath.SS/2
Clath.MS
[1] 2003.536
Clath.F= Clath.MS/22.87
Clath.F
[1] 87.60542
Clath.p=1-pf(Clath.F,2,33)
Clath.p
[1] 6.306067e-14
Clath.Total.SS=sum(A.Clathrates[,2])
Clath.Total.SS
[1] 4761.816
```

	df	SS	MS	F	p-value
Reg	2	4007.072	2003.536	87.60542	6.306067e-14
Res	33	754.74	22.87		
Total	35	4761.816			

We find that the model is significant, given the very small p-value.

c.

Using t-tests, determine the contribution of each regressor to the model. We test this by calculating t-statistics for B1:B2.

```
summary(Clathrates.fit)

Call:
lm(formula = Clath ~ Surfactant + Time)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-9.7716 -4.1656  0.0802  3.8323  8.3349

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.109e+01  1.669e+00   6.642 1.48e-07 ***
Surfactant  3.501e+02  3.968e+01   8.823 3.38e-10 ***
Time        1.089e-01  9.983e-03  10.912 1.74e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.782 on 33 degrees of freedom
Multiple R-squared:  0.8415,   Adjusted R-squared:  0.8319
F-statistic:  87.6 on 2 and 33 DF,  p-value: 6.316e-14
Ho: Beta1=0
Surfactant  3.501e+02  3.968e+01
t_B1=(3.501e+02-0)/ 3.968e+01
t_B1
[1] 8.823085
p_B1=2*(pt(-abs(t_B1),33))
p_B1
[1] 3.377245e-10



Ho: Beta2=0
Time        1.089e-01  9.983e-03
t_B2=(1.089e-01-0)/ 9.983e-03
t_B2
[1] 10.90854
p_B2=2*(pt(-abs(t_B2),33))
p_B2
[1] 1.755484e-12
```

We conclude that both B2 and B3 contribute to the model.

d.

Calculate R^2 and RAdj^2 and compare these to the R^2 and RAdj^2 for a reduced model comparing clathrate formation to time (x2).

```
R_2=Clath.SS/Clath.Total.SS
R_2
[1] 0.8415008

R_2_Adj=1-(1-R_2)*(33/35)
R_2_Adj
[1] 0.8505579


Clath.fit.reduced =lm(Clath~ Time)
A.Clath.reduced=anova(Clath.fit.reduced)
```

```
Clath.SS.reduced=A.Clath.reduced[1,2]
Clath.Total.SS.reduced= sum(A.Clath.reduced[,2])


summary(Clath.fit.reduced)


Call:
lm(formula = Clath ~ Time)

Residuals:
    Min      1Q  Median      3Q     Max
-12.226  -5.282  -2.261   1.788  19.526

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 18.74945    2.57391   7.284 1.95e-08 ***
Time         0.09770    0.01788   5.465 4.28e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.635 on 34 degrees of freedom
Multiple R-squared:  0.4676,   Adjusted R-squared:  0.452
F-statistic: 29.86 on 1 and 34 DF,  p-value: 4.276e-06



R_2.reduced=Clath.SS.reduced / Clath.Total.SS.reduced
R_2.reduced
[1] 0.467616

R_2_Adj.reduced =1-(1-R_2.reduced)*(33/35)
R_2_Adj.reduced
1] 0.498038…. Or 0.452  by the summary of the fit above
```

There is a large difference between the R^2 values in the full and reduced model. So, we could not use the reduced model instead.

e.

Construct a 95% CI on time (x2) for the full and reduced models.

Full model:

```
Time        1.089e-01  9.983e-03
a=qt(.95,33)
> a
[1] 1.69236
B2hat=1.089e-01
B2_sterr=9.983e-03
Lower_Bound=B2hat-a*B2_sterr
```

```
Lower_Bound
[1] 0.09200517
Upper_Bound=B2hat+a*B2_sterr
Upper_Bound
[1] 0.1257948


Reduced model:

Time          0.09770    0.01788
a=qt(.95,33)
> a
[1] 1.69236
B2hat=0.09770
B2_sterr=0.01788
Lower_Bound=B2hat-a*B2_sterr
Lower_Bound
[1] 0.0674406
Upper_Bound=B2hat+a*B2_sterr
Upper_Bound
[1] 0.1279594


The confidence interval for the reduced model is slightly wider.
```

Problem 3.17

A chemical engineer investigates how the conversion of raw material (y) depends on reaction temperature (x1) and reaction time (x2). He developed the following models:

Y= 100 + .2 x1 + 4 x2

Y= 95 + .15 x1 + 3 x2 + 1 x1x2

Range: 20<x1<50 .5<x2<10

a.

What is the predicted value of conversion when x2=2 in terms of x1? And for x2=8? Draw a graph of predicted values as a function of temp for both models. What's the effect of the interaction term?

x2=2

```
Y= 100 + .2 x1 + 4 x2
 = 100 + .2 x1 + 4 *2
 = 100 + .2 x1 + 8
 = 108 + .2 x1
```

Y= 95 + .15 x1 + 3 x2 + 1 x1x2
 = 95 + .15 x1 + 3 *2 + 1 x1*2
 = 95 + .15 x1 + 6 + 2 x1
 = 101 + 2.15 x1

x2=8

Y= 100 + .2 x1 + 4 x2
 = 100 + .2 x1 + 4 *8
 = 100 + .2 x1 + 32
 = 132 + .2 x1

Y= 95 + .15 x1 + 3 x2 + 1 x1x2
 = 95 + .15 x1 + 3 *8 + 1 x1*8
 = 95 + .15 x1 + 24 + 8 x1
 = 119 + 8.15 x1

In the first model the value of x2 affects the intercept, while in the second model it affects both the intercept and the slope.

b.

Find the expected change in mean conversion for a unit change in temperature x1 for model 1 when x2=5. Does this quantity depend on the reaction time value?

x2=5

Y= 100 + .2 x1 + 4 x2
 = 100 + .2 x1 + 4 *5
 = 100 + .2 x1 + 20
 = 120 + .2 x1

This quantity does not depend on the expected change in mean conversion for a unit change in temperature. It is .2 and does not depend on the value of x2.

c.

Find the expected change in mean conversion for a unit change in temperature x1 for model 2 when x2=5. Repeat this calculation for x2=2 and x2=8. Does the result depend on the value selected for x2?

x2=5

Y= 95 + .15 x1 + 3 x2 + 1 x1x2

```
 = 95  + .15 x1 + 3 *5 + 1 x1*5
 = 95  + .15 x1 + 15 + 5 x1
 = 110  + 5.15 x1
```

The expected change in mean conversion for a unit change in temperature is 5.15. This does depend on x2.

x2=2

```
Y= 95  + .15 x1 + 3 x2 + 1 x1x2
 = 95  + .15 x1 + 3 *2 + 1 x1*2
 = 95  + .15 x1 + 6 + 2 x1
 = 101  + 2.15 x1
```

x2=8

```
Y= 95  + .15 x1 + 3 x2 + 1 x1x2
 = 95  + .15 x1 + 3 *8 + 1 x1*8
 = 95  + .15 x1 + 24 + 8 x1
 = 119  + 8.15 x1
```

The expected change in mean conversion for a unit change in temperature is 2.15 when x2=2 and is 8.15 when x2=8. In general, the expected change in mean conversion for a unit change in temperature is .15+x2.

Problem 4.2

Consider the multiple regression model for the NFL data.

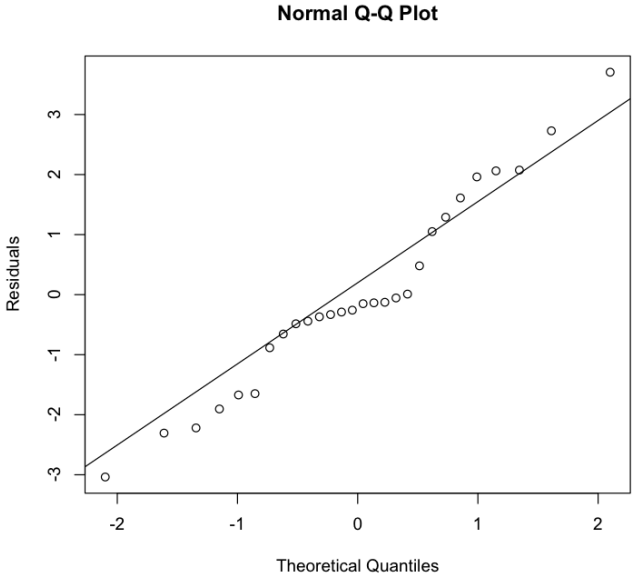
a.

Construct a normal probability plot of the residuals. Does there seem to be any problem with the normality assumption?

```
NFL=read.csv("data-table-B1.csv")
Games_won=NFL$y
Passing_yardage=NFL$x2
Percent_rush_plays=NFL$x7
Yards_rush=NFL$x8
```

```
R.data=data.frame(Games_won, Passing_yardage, Percent_rush_plays, Yards_rush)
NFL.fit=lm(Games_won ~.,R.data)
NFL.residuals=residuals(NFL.fit)
```

```
NFL.fitted=fitted(NFL.fit)

qqnorm(NFL.residuals ,ylab="Residuals")
qqline(NFL.residuals)
```

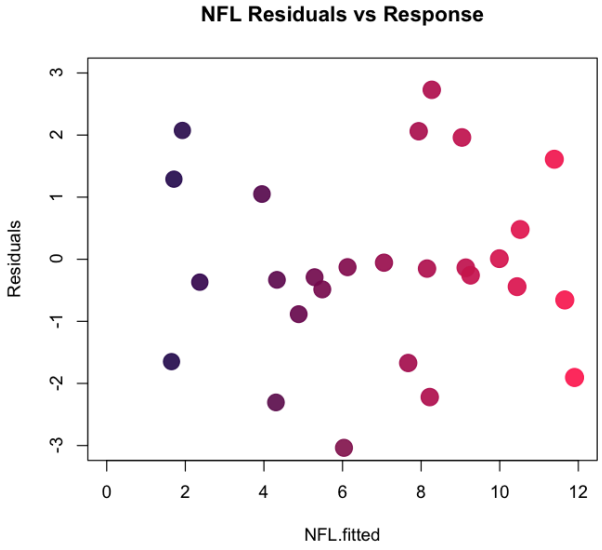


There is a bit more data in the middle than on the edges, but not much more. So, our data might not be normal. There is no overall pattern other than that.

b.

Construct and interpret a plot of the residuals versus the predicted response.

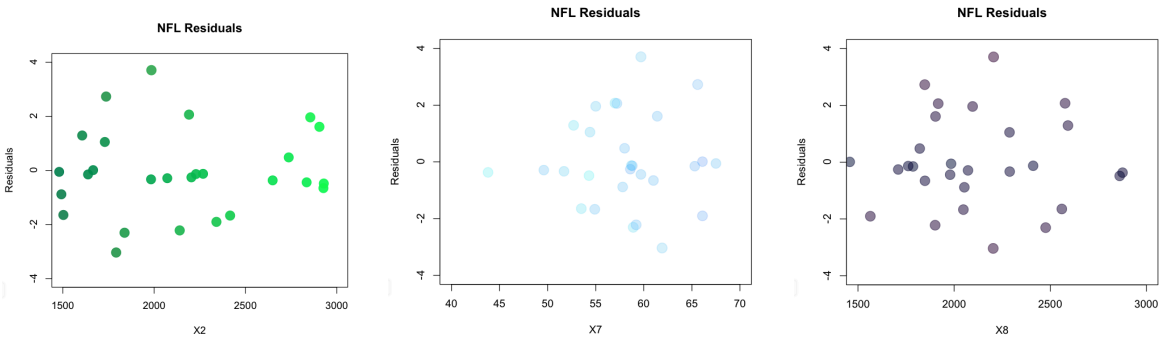
```
y=abs(NFL.residuals)
x=abs(NFL.fitted)
plot(NFL.fitted, NFL.residuals , xlim=c(0,12) , ylim=c(-3,3) , pch=20 , bg="white" , cex=3+(x/30)
, col=rgb(x/12,y/40,0.3,0.9) , main="NFL Residuals vs Response",xlab="NFL.fitted",ylab="
Residuals ")
```



c.

Construct plots of the residuals versus each of the regressor variables. Do these plots imply that the regressor is correctly specified?

```
a=abs(NFL.residuals)
b=abs(Passing_yardage)
c=abs(Percent_rush_plays)
d=abs(Yards_rush)
plot(Passing_yardage, NFL.residuals , xlim=c(1500,3000) , ylim=c(-4,4) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/20,b/3000,0.3,0.9) , main="NFL Residuals",xlab="X2",ylab=" Residuals ")
plot(Percent_rush_plays, NFL.residuals , xlim=c(40,70) , ylim=c(-4,4) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/20,d/3000,0.99,0.2) , main="NFL Residuals",xlab="X7",ylab=" Residuals ")
plot(Yards_rush, NFL.residuals , xlim=c(1500,3000) , ylim=c(-4,4) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/20,c/3000,0.2,0.5) , main="NFL Residuals",xlab="X8",ylab=" Residuals ")
```



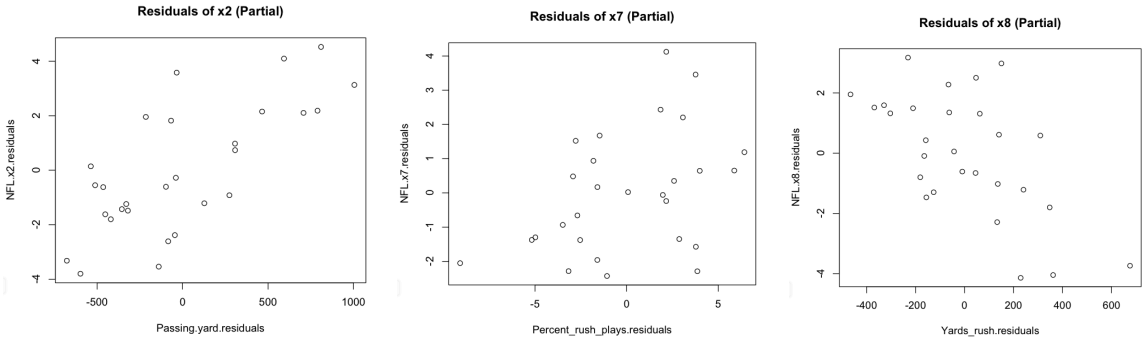
Passing_yardage : green
Percent_rush_plays: baby blue
Yards_rush : dark plum

The baby blue residuals are for the percent of rushing plays and these increase in spread to the right, so they might not be independent and identically distributed (iid). They show nonconstant variance.

d.

Construct the partial regression plots for this model. Compare the plots with the plots of residuals versus regressors from part (c) above. Discuss the type of information provided by these plots.

```
NFL.x2.fit=lm(Games_won~.-Passing_yardage, R.data)
Passing.yard.fit=lm(Passing_yardage~Percent_rush_plays+ Yards_rush,NFL)
NFL.x2.residuals=residuals(NFL.x2.fit)
Passing.yard.residuals=residuals(Passing.yard.fit)
plot(Passing.yard.residuals ,NFL.x2.residuals, main="Residuals of x2
(Partial)")

NFL.x7.fit=lm(Games_won~.- Percent_rush_plays, R.data)
Percent_rush_plays.fit=lm(Percent_rush_plays ~ Passing_yardage +
Yards_rush,NFL)
NFL.x7.residuals=residuals(NFL.x7.fit)
Percent_rush_plays.residuals=residuals(Percent_rush_plays.fit)
plot(Percent_rush_plays.residuals, NFL.x7.residuals, main="Residuals of x7
(Partial)")

NFL.x8.fit=lm(Games_won~.-Yards_rush, R.data)
Yards_rush.fit=lm(Yards_rush ~ Passing_yardage + Percent_rush_plays,NFL)
NFL.x8.residuals=residuals(NFL.x8.fit)
Yards_rush.residuals=residuals(Yards_rush.fit)
plot(Yards_rush.residuals ,NFL.x8.residuals,  main="Residuals of x8
(Partial)")
```



e.

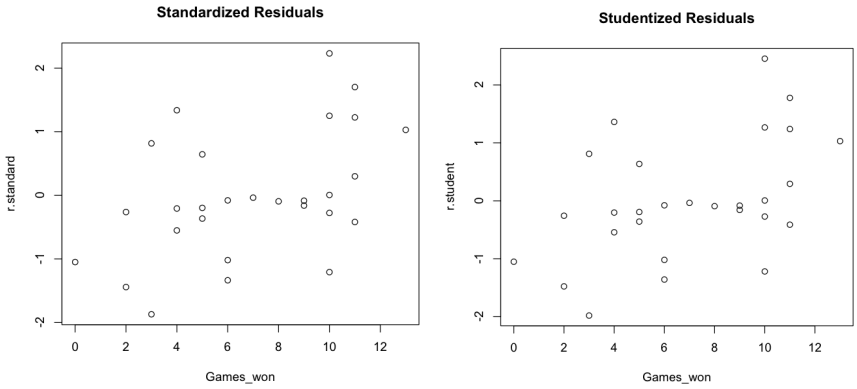
Compute the studentized residuals and the Rstudent residuals for this model. What information is conveyed by these scaled residuals?

```
library(MASS)
r.standard=rstandard(NFL.fit)
r.student=rstudent(NFL.fit)
r.difference=r.standard-r.student
```

```
> r.standard
          1           2           3           4           5
 2.231851618  1.225616368  1.702625305  1.029767789  0.006124483
          6           7           8           9          10
-0.418876221 -1.206836995  0.299328499  1.338032316 -1.441760607
         11          12          13          14          15
-0.036468456  1.251090093 -0.083851688 -0.160668820 -1.335367350
         16          17          18          19          20
 0.644990078 -0.196937383 -0.365011749 -0.078998342 -0.206464327
         21          22          23          24          25
-1.869940122  0.817274105 -0.551056514 -0.276544687 -1.018586104
         26          27          28
-0.094055761 -0.262130195 -1.048746774
> r.student
          1           2           3           4           5
 2.454354223  1.239218310  1.777586702  1.031123075  0.005995537
          6           7           8           9          10
-0.411563960 -1.218993620  0.293574644  1.361631132 -1.476806719
         11          12          13          14          15
-0.035701602  1.266752172 -0.082098218 -0.157370596 -1.358701256
         16          17          18          19          20
 0.636954384 -0.192946834 -0.358322410 -0.077345090 -0.202296957
         21          22          23          24          25
-1.980521136  0.811437522 -0.542899513 -0.271154408 -1.019417881
         26          27          28
-0.092092392 -0.256979177 -1.051031132
```

The first data point might be an outlier.

```
plot(Games_won,r.standard, main="Standardized Residuals")
plot(Games_won,r.student, main="Studentized Residuals")
```

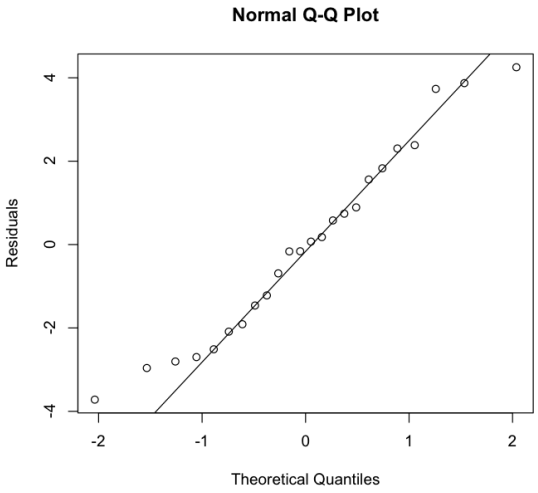


Problem 4.5

Consider the multiple regression model for house price data from 3.7.

a.
Construct a normal probability plot of the residuals. Does there seem to be any problem with the normality assumption?

```
House=read.csv("data-table-B4.csv")
Y=House$y
X1=House$x1
X2=House$x2
X3=House$x3
X4=House$x4
X5=House$x5
X6=House$x6
X7=House$x7
X8=House$x8
X9=House$x9



R.data=data.frame(Y,X1,X2,X3,X4,X5,X6,X7,X8,X9)
House.fit=lm(Y ~.,R.data)
House.residuals=residuals(House.fit)
House.fitted=fitted(House.fit)

qqnorm(House.residuals ,ylab="Residuals")
qqline(House.residuals)
```

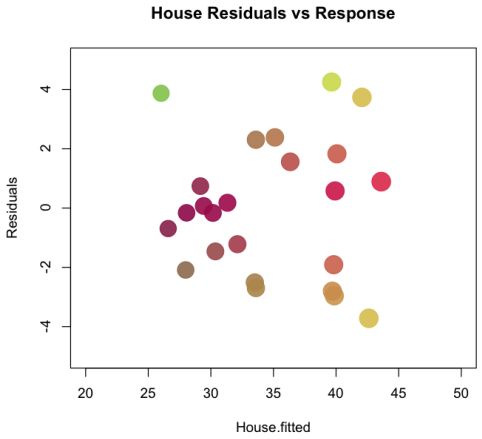


There is no overall pattern. So, our data is probably normal.

b.

Construct and interpret a plot of the residuals versus the predicted response.

```
y=abs(House.residuals)
x=abs(House.fitted)
plot(House.fitted, House.residuals , xlim=c(20,50) , ylim=c(-5,5) , pch=20 , bg="white" ,
cex=3+(x/30) , col=rgb(x/50,y/5,0.3,0.9) , main="House Residuals vs
Response",xlab="House.fitted",ylab=" Residuals ")
```



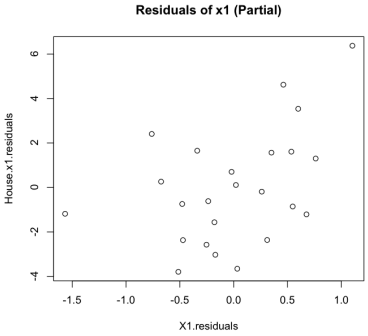
We see a very slight upward drift.

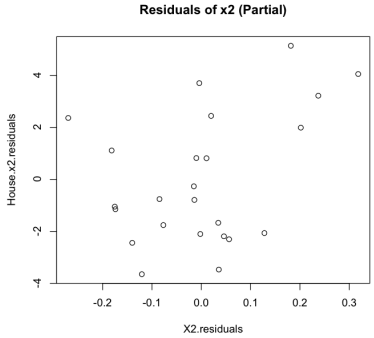
c.

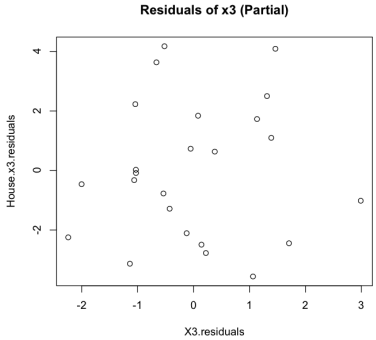
Construct the partial regression plots for this model. Does it seem some variables not necessary?

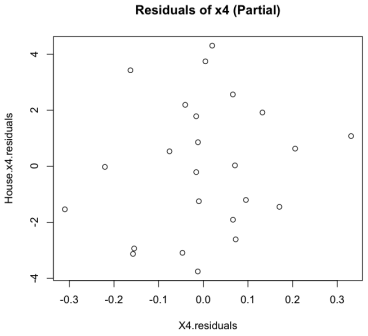
```
R.data.x=data.frame(X1, X2,X3,X4,X5,X6,X7,X8,X9)

House.x1.fit=lm(Y~.-X1, R.data)
X1.fit=lm(X1~.,R.data.x)
House.x1.residuals=residuals(House.x1.fit)
X1.residuals=residuals(X1.fit)
plot(X1.residuals, House.x1.residuals, main="Residuals of x1 (Partial)")

House.x2.fit=lm(Y~.-X2, R.data)
X2.fit=lm(X2~.,R.data.x)
House.x2.residuals=residuals(House.x2.fit)
X2.residuals=residuals(X2.fit)
plot(X2.residuals, House.x2.residuals, main="Residuals of x2 (Partial)")

House.x3.fit=lm(Y~.-X3, R.data)
X3.fit=lm(X3~.,R.data.x)
House.x3.residuals=residuals(House.x3.fit)
X3.residuals=residuals(X3.fit)
plot(X3.residuals, House.x3.residuals, main="Residuals of x3 (Partial)")

House.x4.fit=lm(Y~.-X4, R.data)
X4.fit=lm(X4~.,R.data.x)
House.x4.residuals=residuals(House.x4.fit)
X4.residuals=residuals(X4.fit)
plot(X4.residuals, House.x4.residuals, main="Residuals of x4 (Partial)")

House.x5.fit=lm(Y~.-X5, R.data)
X5.fit=lm(X5~.,R.data.x)
House.x5.residuals=residuals(House.x5.fit)
X5.residuals=residuals(X5.fit)
plot(X5.residuals, House.x5.residuals, main="Residuals of x5 (Partial)")

House.x6.fit=lm(Y~.-X6, R.data)
X6.fit=lm(X6~.,R.data.x)
House.x6.residuals=residuals(House.x6.fit)
X6.residuals=residuals(X6.fit)
plot(X6.residuals, House.x6.residuals, main="Residuals of x6 (Partial)")

House.x7.fit=lm(Y~.-X7, R.data)
```

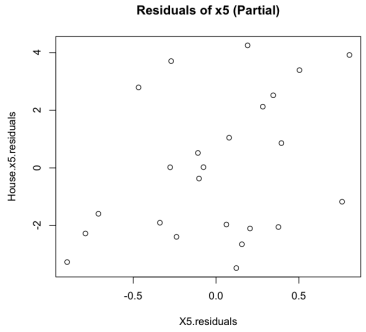
```
X7.fit=lm(X7~.,R.data.x)
House.x7.residuals=residuals(House.x7.fit)
X7.residuals=residuals(X7.fit)
plot(X7.residuals, House.x7.residuals, main="Residuals of x7 (Partial)")

House.x8.fit=lm(Y~.-X8, R.data)
X8.fit=lm(X8~.,R.data.x)
House.x8.residuals=residuals(House.x8.fit)
X8.residuals=residuals(X8.fit)
plot(X8.residuals, House.x8.residuals, main="Residuals of x8 (Partial)")

House.x9.fit=lm(Y~.-X9, R.data)
X9.fit=lm(X9~.,R.data.x)
House.x9.residuals=residuals(House.x9.fit)
X9.residuals=residuals(X9.fit)
plot(X9.residuals, House.x9.residuals, main="Residuals of x9 (Partial)")
```

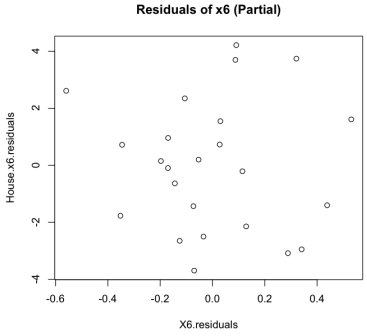


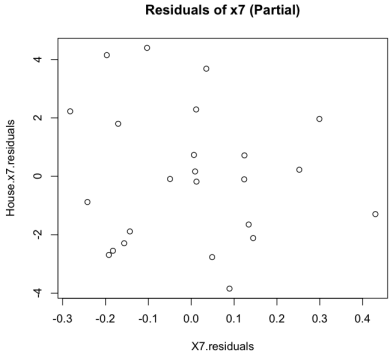


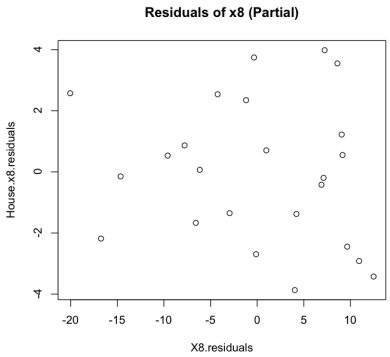


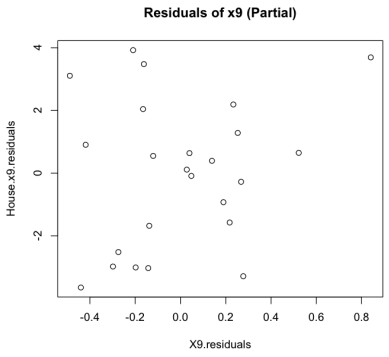












Most of these don't show a strong linear relationship. X1 shows the strongest one.

d.

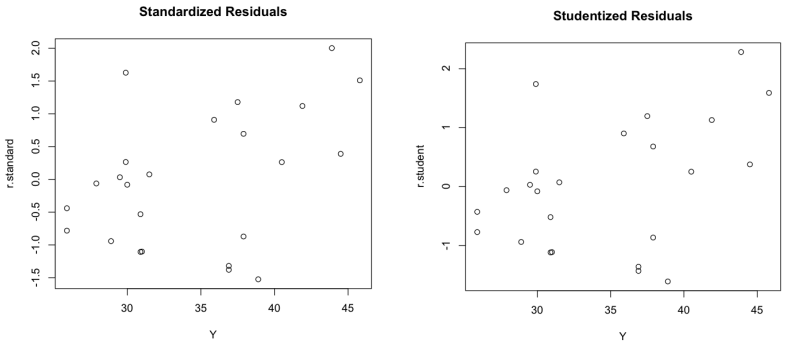
Compute the studentized residuals and the Rstudent residuals for this model. What information is conveyed by these scaled residuals?

```
library(MASS)
r.standard=rstandard(House.fit)
r.student=rstudent(House.fit)
r.difference=r.standard-r.student
```

```
> r.standard
          1           2           3           4           5
 0.03216502 -0.06277433 -0.78261093  0.26457412  1.62698920
          6           7           8           9          10
-0.53192148 -0.94268410  0.90861299  0.07578361 -1.10224799
         11          12          13          14          15
-1.10687729 -0.08211665 -1.32022301  1.11933252  0.26271508
         16          17          18          19          20
 2.00247153  1.17868251 -0.87205543  0.38936776  0.69461334
         21          22          23          24
-1.52506014 -1.37994057  1.51056017 -0.44091589
> r.student
          1           2           3           4           5
 0.03099613 -0.06049938 -0.77120089  0.25558977  1.74101624
          6           7           8           9          10
-0.51783177 -0.93867275  0.90257816  0.07304190 -1.11147523
         11          12          13          14          15
-1.11658995 -0.07914865 -1.35964861  1.13038189  0.25378493
         16          17          18          19          20
 2.28429896  1.19673719 -0.86413114  0.37725238  0.68118709
         21          22          23          24
-1.60933049 -1.43059072  1.59103270 -0.42785823
```

Data point number 16 might be an outlier.

```
plot(Y,r.standard, main="Standardized Residuals")
plot(Y,r.student, main="Studentized Residuals")
```



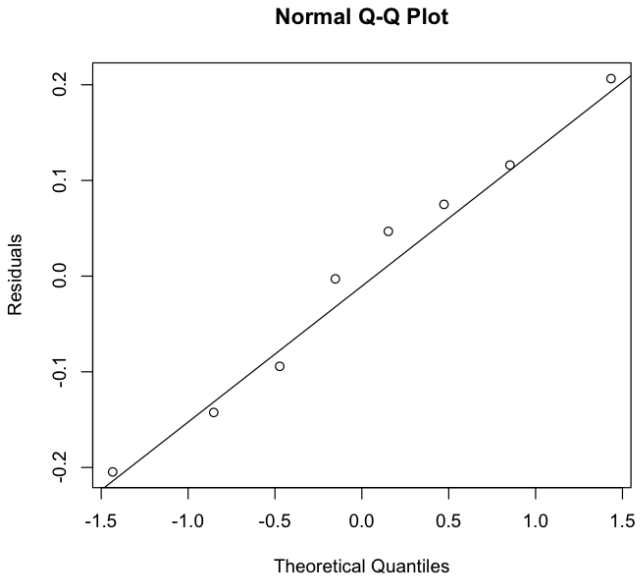
Problem 4.10

Consider the multiple regression model for viscosity data from 2.14.

a.

Construct a normal probability plot of the residuals. Does there seem to be any problem with the normality assumption?

```
Viscosity=read.csv("Viscosity_data.csv")
Y=Viscosity$Viscosity
X=Viscosity$Ratio


R.data=data.frame(Y,X)
Viscosity.fit=lm(Y ~.,R.data)
Viscosity.residuals=residuals(Viscosity.fit)
Viscosity.fitted=fitted(Viscosity.fit)

qqnorm(Viscosity.residuals ,ylab="Residuals")
qqline(Viscosity.residuals)
```

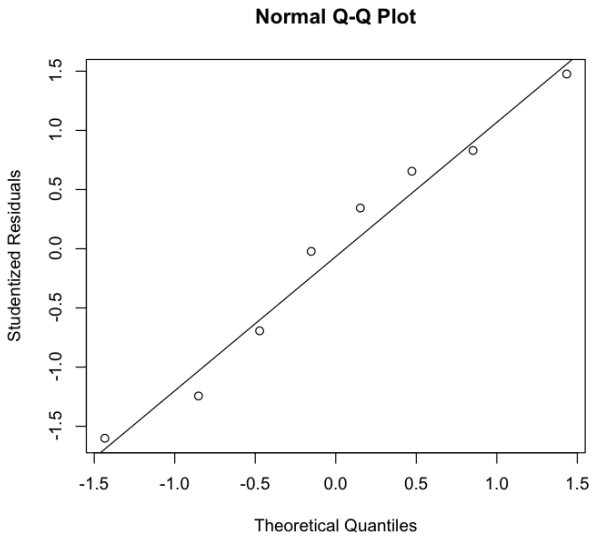


There is no overall pattern. So, our data is probably normal.

b.

Repeat part a using studentized residuals. Is there a substantial difference?

```
library(MASS)
r.standard=rstandard(Viscosity.fit)
qqnorm(r.standard,ylab="Studentized Residuals")
qqline(r.standard)
```

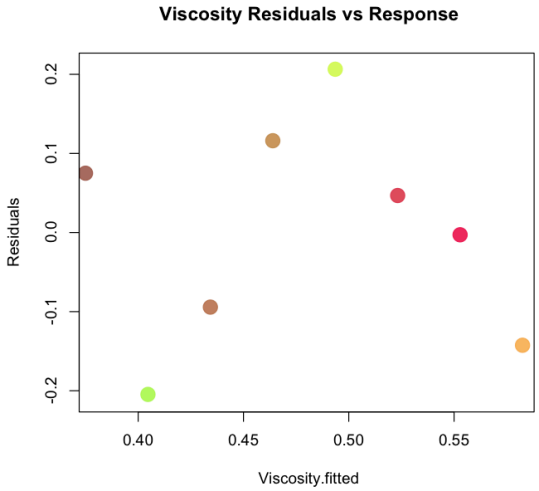


The pattern is identical, but the scale is different. The studentized resisuals range: (-1.5,1.5), while the regular residuals range: (-.2,.2).

c.

Construct a plot of the residuals vs predicted response.

```
y=abs(Viscosity.residuals)
x=abs(Viscosity.fitted)
plot(Viscosity.fitted, Viscosity.residuals , xlim=c(.38,.58) , ylim=c(-.21,.21) , pch=20 ,
bg="white" , cex=3+(x/30) , col=rgb(x/.6,y/.21,0.3,0.9) , main=" Viscosity Residuals vs
Response",xlab=" Viscosity.fitted",ylab=" Residuals ")
```



There might be a right-rotated S shape pattern, but it is hard to tell because there are so few data points.

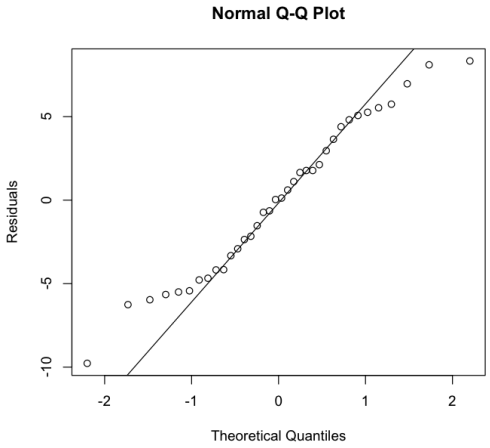
Problem 4.16

Consider the multiple regression model for clathrate formation from 3.12.

a.

Construct a normal probability plot of the residuals. Does there seem to be any problem with the normality assumption?

```
Clathrates=read.csv("data-table-B8.csv")
Clath=Clathrates$y
Surfactant= Clathrates $x1
Time= Clathrates $x2


R.data=data.frame(Clath,Surfactant,Time)
Clath.fit=lm(Clath ~.,R.data)
Clath.residuals=residuals(Clath.fit)
Clath.fitted=fitted(Clath.fit)

qqnorm(Clath.residuals ,ylab="Residuals")
qqline(Clath.residuals)
```

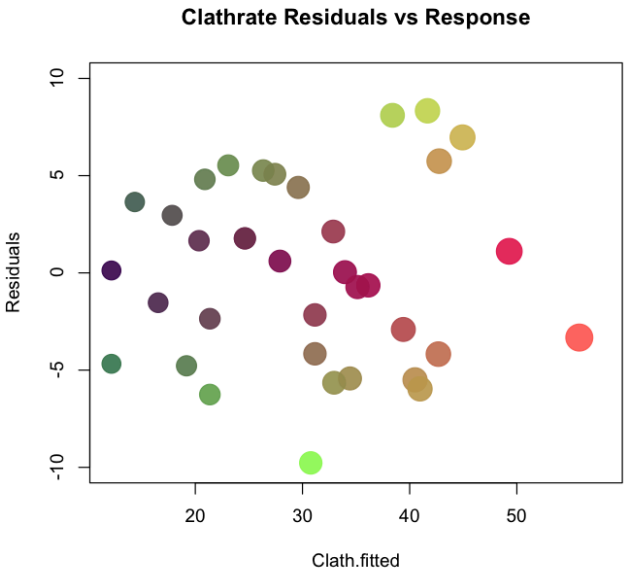


The tails drift off to the side. So, the tails might not fit a normal distribution.

b.

Construct and interpret a plot of the residuals versus the predicted response.

```
y=abs(Clath.residuals)
x=abs(Clath.fitted)
plot(Clath.fitted, Clath.residuals , xlim=c(12,58) , ylim=c(-10,10) , pch=20 , bg="white" ,
cex=3+(x/30) , col=rgb(x/56,y/10,0.3,0.9) , main="Clathrate Residuals vs
Response",xlab="Clath.fitted",ylab=" Residuals ")
```



Maybe a slight upward drift, but maybe not. Looks Good.

c.

With the reduced model from 3.12- Compute a PRESS statistic for both models. Which is better?

```
install.packages("MPV")
library(MPV)

Clath.fit=lm(Clath ~.,R.data)
Clath.fit.reduced=lm(Clath ~Time,R.data)

PRESS(Clath.fit)
[1] 916.4096

PRESS(Clath.fit.reduced)
[1] 2825.624
```

We want a smaller PRESS value. So, we conclude that the full model is a better predictor of clathrate formation.

Problem 4.22

Table B14 contains data on transient points of an electronic inverter. Using only regressors x1:x4 fit a multiple regression model to the data

a.
Investigate the adequacy of the model.

```
Inverter=read.csv("data-table-B14.csv")
Y= Inverter$y
X1= Inverter$x1
X2= Inverter$x2
X3= Inverter$x3
X4= Inverter$x4

R.data=data.frame(Y,X1,X2,X3,X4)
Inverter.fit=lm(Y ~.,R.data)
```

```
> Inverter.fit

Call:
lm(formula = Y ~ ., data = R.data)

Coefficients:
(Intercept)           X1           X2           X3           X4
     3.1482      -0.2900       0.1992       0.4554      -0.6092
```

$$y = 3.1482 - 0.2900\,x_1 + 0.1992\,x_2 + 0.4554\,x_3 - 0.6092\,x_4$$

```
A.Inverter=anova(Inverter.fit)
> A.Inverter
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value   Pr(>F)
X1         1 13.045  13.045  2.8389 0.107548
X2         1 47.708  47.708 10.3825 0.004274 **
X3         1 11.871  11.871  2.5835 0.123658
X4         1 42.879  42.879  9.3316 0.006254 **
Residuals 20 91.901   4.595
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
Inv.SS=sum(A.Inverter[1:4,2])
Inv.SS
[1] 115.5029
```

```
Inv.MS=Inv.SS/4
Inv.MS
[1] 28.87572
Inv.F=Inv.MS/4.595
Inv.F
[1] 6.284161
Inv.p=1-pf(Inv.F,4,20)
Inv.p
[1] 0.001916493
Inv.SS.Total=sum(A.Inverter[,2])
Inv.SS.Total
[1] 207.4037
```

	df	SS	MS	F	p-value
Reg	5	115.5029	28.87572	6.284161	0.001916493
Res	20	91.901	4.595		
Total	25	207.4037			

We conclude that our model is significant based on the small p-value.

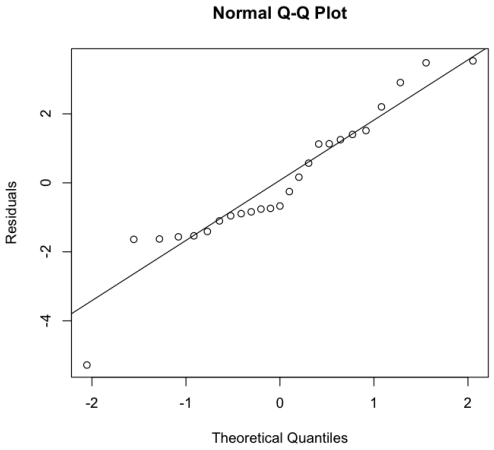
```
> summary(Inverter.fit)

Call:
lm(formula = Y ~ ., data = R.data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2806 -1.1030 -0.6715  1.2499  3.5333

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.14825    1.43891   2.188  0.04072 *
X1          -0.28999    0.11463  -2.530  0.01992 *
X2           0.19919    0.06891   2.891  0.00904 **
X3           0.45537    0.18321   2.486  0.02190 *
X4          -0.60919    0.19942  -3.055  0.00625 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.144 on 20 degrees of freedom
Multiple R-squared:  0.5569,   Adjusted R-squared:  0.4683
F-statistic: 6.284 on 4 and 20 DF,  p-value: 0.001917
```

These values could be higher →

```
R-squared:  0.5569
Adjusted R-squared:  0.4683
```

We check our data for normality.

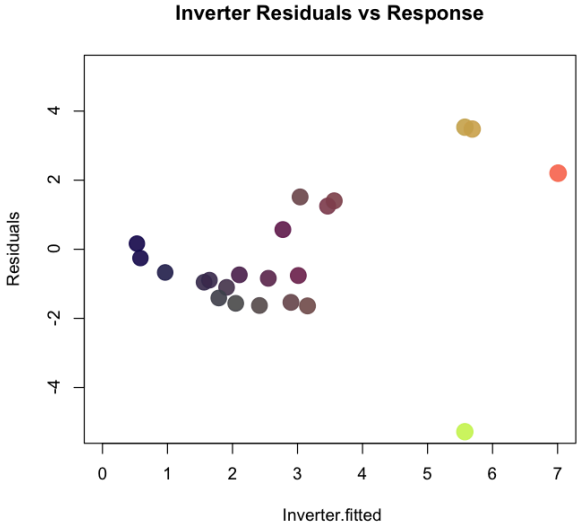
```
Inverter.residuals=residuals(Inverter.fit)
Inverter.fitted=fitted(Inverter.fit)

qqnorm(Inverter.residuals ,ylab="Residuals")
qqline(Inverter.residuals)
```



The left tail drifts off to the side. Otherwise, looks pretty normal.

We consider the relationship between residuals and the predicted response.

```
y=abs(Inverter.residuals)
x=abs(Inverter.fitted)
plot(Inverter.fitted, Inverter.residuals , xlim=c(0,7) , ylim=c(-5.2,5.2) , pch=20 , bg="white" ,
cex=3+(x/30) , col=rgb(x/7.2,y/5.5,0.3,0.9) , main="Inverter Residuals vs Response",xlab="
Inverter.fitted",ylab=" Residuals ")
```



There is an upward drift.

We use PRESS statistics to compare our full model to a reduced model consisting of all regressors, except one. We call the reduced model without X1 “reduced.x1.”

```
R.data=data.frame(Y,X1,X2,X3,X4)
Inverter.fit=lm(Y ~.,R.data)
Inverter.fit.reduced.x1=lm(Y ~X2+X3+X4,R.data)
Inverter.fit.reduced.x2=lm(Y ~X1+X3+X4,R.data)
Inverter.fit.reduced.x3=lm(Y ~X1+X2+X4,R.data)
Inverter.fit.reduced.x4=lm(Y ~X1+X2+X3,R.data)

install.packages("MPV")
library(MPV)


PRESS(Inverter.fit)
[1] 238.2421
PRESS(Inverter.fit.reduced.x1)
[1] 264.9785
PRESS(Inverter.fit.reduced.x2)
[1] 179.8409
PRESS(Inverter.fit.reduced.x3)
[1] 223.2453
PRESS(Inverter.fit.reduced.x4)
[1] 282.5171
```

The lowest PRESS value is from the reduced model that does not include x2. The PRESS value for the reduced model that does not include x3 is lower than the PRESS value for the full model. Reanalyzing the model without x2 sounds like a good idea. Maybe removing x3 would also be a good idea.

```
library(MASS)
r.standard=rstandard(Inverter.fit)

> r.standard
          1           2           3           4           5
-0.80923944 -3.37070327 -0.40824654  2.03600400 -0.53244458
          6           7           8           9          10
 0.63814536 -0.46216112  2.02235626  1.34266426 -0.43777320
         11          12          13          14          15
 0.76259915 -0.32802988 -0.37790473  1.80017071  0.08361776
         16          17          18          19          20
-0.70818694 -0.80207185  0.28507485  0.60970743  0.60593475
         21          22          23          24          25
-0.12559984  0.72782473 -0.74097592 -0.83280584 -0.46448157
```

It looks like the second observation is an outlier

b.

Suppose observation 2 was recorded incorrectly. Delete this observation and refit the model. Perform a thorough residual analysis.

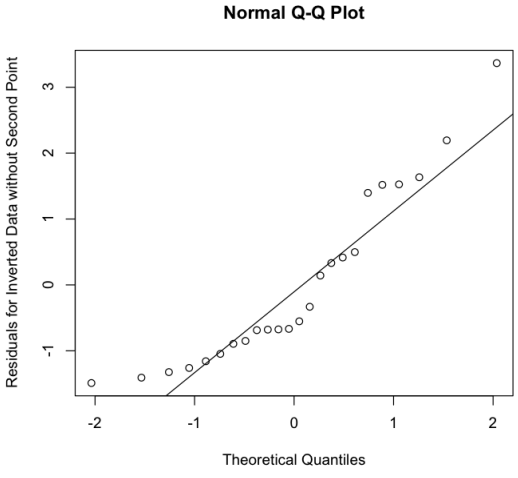
For this problem I deleted the row in the original Excel file with this data table and resaved it as a csv, calling it "data-table-B14_minus_obs_2."

```
Inverter_minus_obs_2=read.csv("data-table-B14_minus_obs_2.csv")
Y_2= Inverter_minus_obs_2$y
X1_2= Inverter_minus_obs_2$x1
X2_2= Inverter_minus_obs_2$x2
X3_2= Inverter_minus_obs_2$x3
X4_2= Inverter_minus_obs_2$x4
```

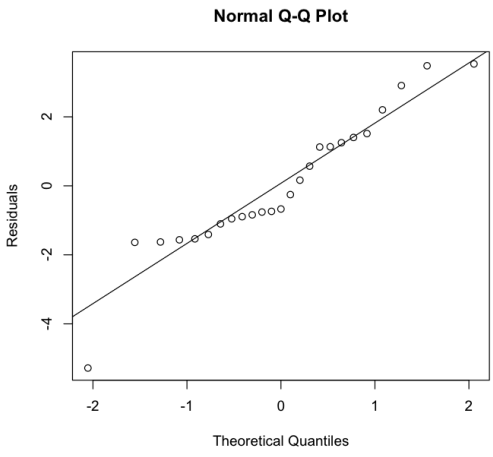
```
R.data_2=data.frame(Y_2,X1_2,X2_2,X3_2,X4_2)
Inverter.fit_2=lm(Y_2 ~.,R.data_2)
```

```
Inverter.residuals_2=residuals(Inverter.fit_2)
Inverter.fitted_2=fitted(Inverter.fit_2)

qqnorm(Inverter.residuals_2 ,ylab="Residuals for Inverted Data without Second
Point")
qqline(Inverter.residuals_2)
```

Without second observation



With second observation

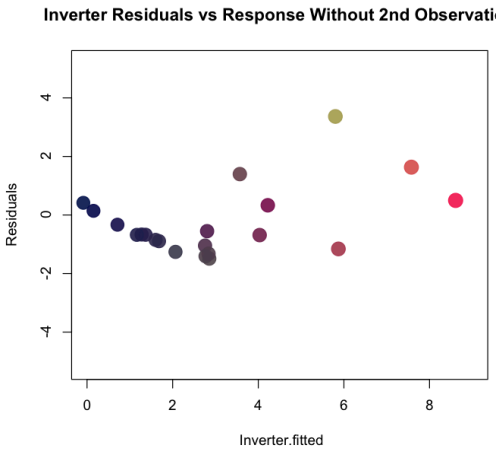


This looks much better. It seems indeed that the second point was an outlier.

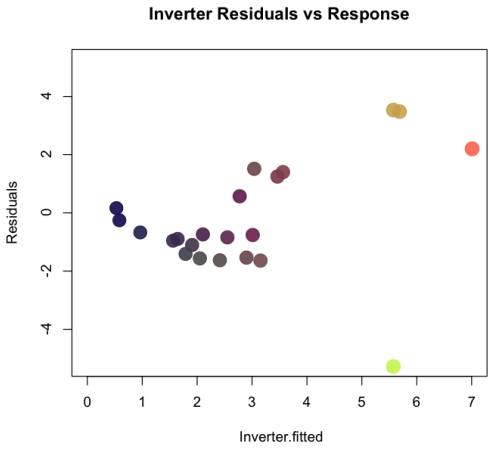
We consider the relationship between residuals and the predicted response for the model with and without the second observation.

```
y=abs(Inverter.residuals_2)
x=abs(Inverter.fitted_2)
plot(Inverter.fitted_2, Inverter.residuals_2 , xlim=c(0,9) , ylim=c(-5.2,5.2) , pch=20 ,
bg="white" , cex=3+(x/30) , col=rgb(x/9,y/5.5,0.3,0.9) , main="Inverter Residuals vs Response
Without 2nd Observation",xlab=" Inverter.fitted",ylab=" Residuals ")
```

Without second observation



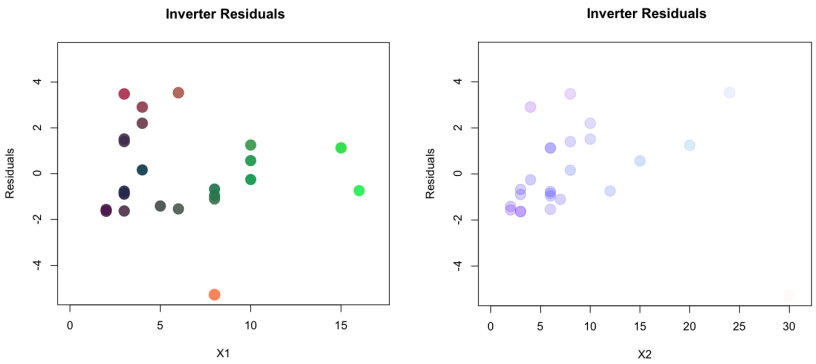
With second observation

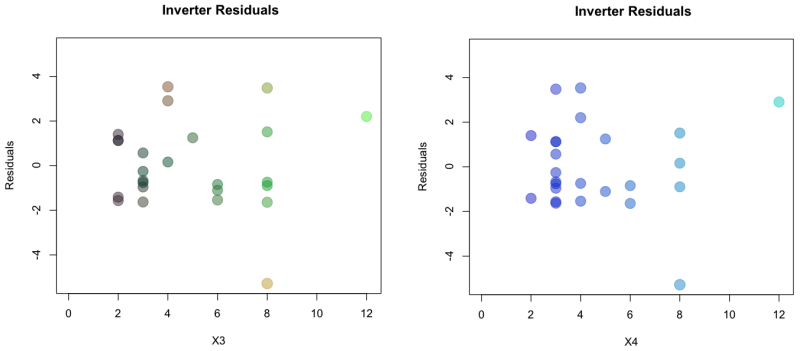


These graphs show that the 2nd data point was adding to the increased spread to the right of the graph.

We construct plots of the residuals versus each of the regressor variables

```
a=abs(Inverter.residuals)
b=abs(X1)
c=abs(X2)
d=abs(X3)
e=abs(X4)
plot(X1, Inverter.residuals , xlim=c(0,17) , ylim=c(-5.3,5.3) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/5.3,b/17,0.3,0.9) , main=" Inverter Residuals",xlab="X1",ylab="
Residuals ")
plot(X2, Inverter.residuals , xlim=c(0,31) , ylim=c(-5.3,5.3) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/5.3,c/31,0.99,0.2) , main=" Inverter Residuals",xlab="X2",ylab="
Residuals ")
plot(X3, Inverter.residuals , xlim=c(0,12.1) , ylim=c(-5.3,5.3) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/7,d/13,0.2,0.5) , main=" Inverter Residuals",xlab="X3",ylab=" Residuals
")
plot(X4, Inverter.residuals , xlim=c(0,12.1) , ylim=c(-5.3,5.3) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/50.3,e/15,0.8,0.5) , main=" Inverter Residuals",xlab="X4",ylab="
Residuals ")
```



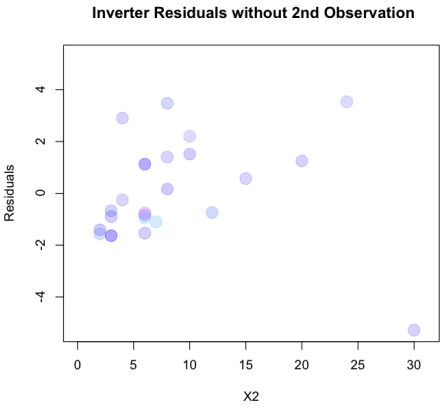


It seems that x2 shows nonconstant variance.

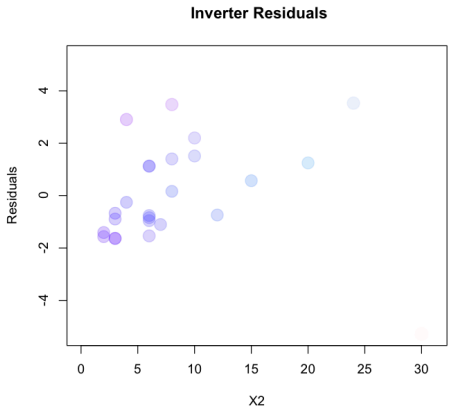
We compare the graph for x2 when we remove the second observation.

```
a=abs(Inverter.residuals_2)
c=abs(X2_2)
plot(X2, Inverter.residuals , xlim=c(0,31) , ylim=c(-5.3,5.3) , pch=20 , bg="white" ,
cex=3+(a/30) , col=rgb(a/5.3,c/31,0.99,0.2) , main=" Inverter Residuals without 2nd
Observation",xlab="X2",ylab=" Residuals ")
```

Without second observation



With second observation



It seems like removing the second observation may have helped with the nonconstant variance in X2 a little bit, but there is still nonconstant variance in this regressor.