Ve406 Lecture 12

Jing Liu

UM-SJTU Joint Institute

June 25, 2018

Yield Study of a chemical process

- The following data were gathered in the course of studying the yield of a particular chemical process.

`yield`	measures the effectiveness of a synthetic procedure
`conversion`	a percentage of desired over undesired product(s)
`flow`	measures the speed of the reaction process
`ratio`	a ratio of between two chemicals

- Chemical theory predicts `yield` is related to the reciprocal of `ratio`.
- Theory also predicts `yield` is related to `conversion*flow`.
- We are interested in the relation between the yield and the other variables.
- In particular, we would like to whether the data support the two theories.

- After loading the data

```
> chem_pro.df = read.table("~/Desktop/chem_pro.csv",
+                           sep = ",", header = TRUE)
```

Q: Do all variables have the right variable type?

```
> str(chem_pro.df)
```

```
'data.frame':   44 obs. of  4 variables:
 $ yield     : num  55.5 54.8 52.2 50.4 49.3 ...
 $ conversion: num  11.8 11.9 12.1 12.1 12 ...
 $ flow      : num  119 105 97 101 44 ...
 $ ratio     : Factor w/ 40 levels "0.036","0.089",..: 16 2 3 6 5 1 7 9 11 24 ...
```

- Somehow `ratio` is a `Factor`.

```
> class(chem_pro.df$ratio)
```

```
[1] "factor"
```

- We look into the values that this factor variable can take

```
> levels(chem_pro.df$ratio)
```

```
 [1] "0.036" "0.089" "0.094" "0.097"
 [5] "0.1"   "0.108" "0.113" "0.116"
 [9] "0.123" "0.126" "0.135" "0.136"
[13] "0.143" "0.152" "0.153" "0.155"
[17] "0.16"  "0.161" "0.164" "0.166"
[21] "0.169" "0.17"  "0.18"  "0.183"
[25] "0.184" "0.188" "0.192" "0.194"
[29] "0.195" "0.197" "0.201" "0.211"
[33] "0.215" "0.221" "0.222" "0.223"
[37] "0.225" "0.229" "0.233" "0>163"
```

- Identifying the observation, which is clearly a typo,

```
> ratio_typo = which(chem_pro.df$ratio == "0>163")
> ratio_typo
```

```
[1] 32
```

- Correcting the typo by first converting it into a character variable

```
> chem_pro.df$ratio = as.character(chem_pro.df$ratio)
```

- Reassigning the value `0.163` instead of `"0>163"`

```
> chem_pro.df$ratio[ratio_typo] = "0.163"
```

- Coercing it into a double precision numeric variable

```
> chem_pro.df$ratio = as.double(chem_pro.df$ratio)
> chem_pro.df$ratio
```

```
 [1] 0.155 0.089 0.094 0.108 0.100 0.036 0.113
 [8] 0.123 0.135 0.183 0.166 0.221 0.192 0.188
[15] 0.201 0.153 0.194 0.097 0.136 0.143 0.116
[22] 0.195 0.160 0.164 0.197 0.233 0.211 0.222
[29] 0.223 0.229 0.170 0.163 0.153 0.180 0.126
[36] 0.152 0.184 0.225 0.169 0.161 0.197 0.201
[43] 0.221 0.215
```

Q: Is there any more obvious error in the dataset?

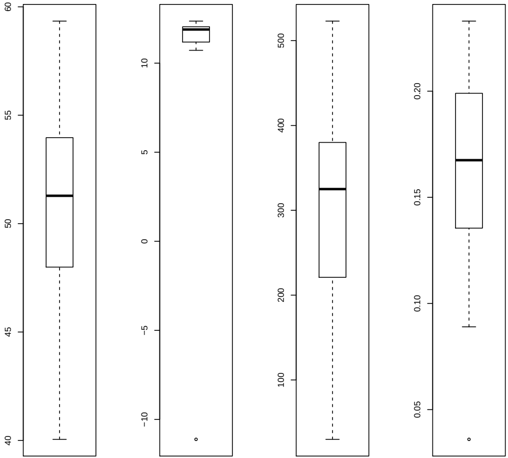
```
> summary(chem_pro.df)
```

```
     yield         conversion          flow            ratio
 Min.   :40.05   Min.   :-11.12   Min.   : 30.0   Min.   :0.0360
 1st Qu.:48.05   1st Qu.: 11.19   1st Qu.:221.5   1st Qu.:0.1358
 Median :51.28   Median : 11.89   Median :325.0   Median :0.1675
 Mean   :50.68   Mean   : 11.11   Mean   :291.9   Mean   :0.1658
 3rd Qu.:53.91   3rd Qu.: 12.04   3rd Qu.:380.0   3rd Qu.:0.1980
 Max.   :59.34   Max.   : 12.36   Max.   :523.0   Max.   :0.2330
```

- Boxplot as well as summary is particularly useful to identify unusual values

```
> par(mfrow = c(1,4))
> lapply(chem_pro.df, boxplot) # do for every column
> par(mfrow = c(1,1))
```

- If there is any clear error, we should correct it if possible.
- Take a note or report if there is any suspicion of error in the data, since every data point matters when there are only 44 of them.



- There seems to be one usually small value in variable 2 and one in variable 4.

- Identifying both of them

```
> names(chem_pro.df)
```

```
[1] "yield"       "conversion" "flow"       "ratio"
```

```
> conversion_typo = which(
+   chem_pro.df$conversion <= -10)
>
> ratio_unusual = which(
+   chem_pro.df$ratio <= 0.05)
>
> conversion_typo; ratio_unusual
```

```
[1] 44
[1] 6
```

- It only makes sense to have `conversion` between 0 and 1, but observation 6 might just be unusual.

- According to the summary,

```
> summary(chem_pro.df$conversion)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -11.12   11.19   11.89   11.11   12.04   12.36
```

it is likely that the minus sign is a typo.

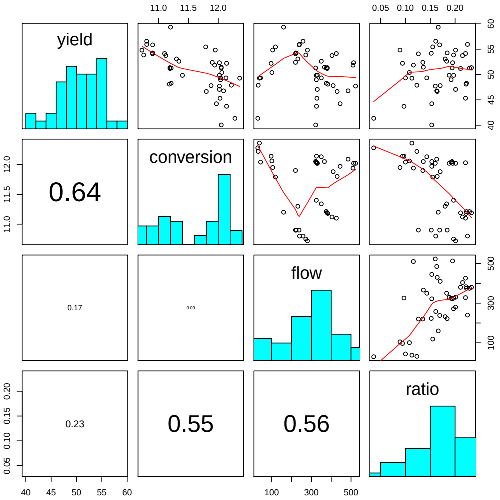
- So we modify the 44th observation and keep record of it

```
> chem_pro.df$conversion[conversion_typo] =
+   - chem_pro.df$conversion[conversion_typo]
```

- We have to keep observation 6 on watch list, it may be a high leverage point.

Q: Is there any unusual point in terms of relationships between variables?

```
> pairs(chem_pro.df,
+       diag.panel = panel.hist,
+       lower.panel = panel.cor,
+       upper.panel = panel.smooth)
```



- There seems to be no more unusual point, and no strong linear relationship.

- We can consider 3d scatter plots, but there seems to no indication of trouble.

```
> library(scatterplot3d)
>
> vname = names(chem_pro.df)
> k = length(vname) - 1
> m = combn(k, 2)
>
> pdf()
> for (j in 1:ncol(m)){
+   scatterplot3d(chem_pro.df[, m[1, j]],
+                 chem_pro.df[, m[2, j]],
+                 chem_pro.df$yield, type="h",
+                 xlab = vname[m[1, j]],
+                 ylab = vname[m[2, j]],
+                 zlab = "yield")
+ }
> dev.off()
```

- We start by fitting the basic model

```
> chem_pro.LM = lm(yield~., data = chem_pro.df)
> summary(chem_pro.LM)
```

```
Call:
lm(formula = yield ~ ., data = chem_pro.df)

Residuals:
    Min      1Q  Median      3Q     Max
-8.1715 -1.3101  0.1171  1.5926  5.5769

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept) 113.011270  14.631935   7.724 1.88e-09 ***
conversion   -5.189435   1.149651  -4.514 5.49e-05 ***
flow         -0.006983   0.004467  -1.563    0.126
ratio        -0.055762  15.755755  -0.004    0.997
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.118 on 40 degrees of freedom
Multiple R-squared:   0.46,     Adjusted R-squared:  0.4195
F-statistic: 11.36 on 3 and 40 DF,  p-value: 1.591e-05
```

- Given all the assumptions were OK, it seems only conversion is contributing to the regression, a low yield would be associated with a high conversion.

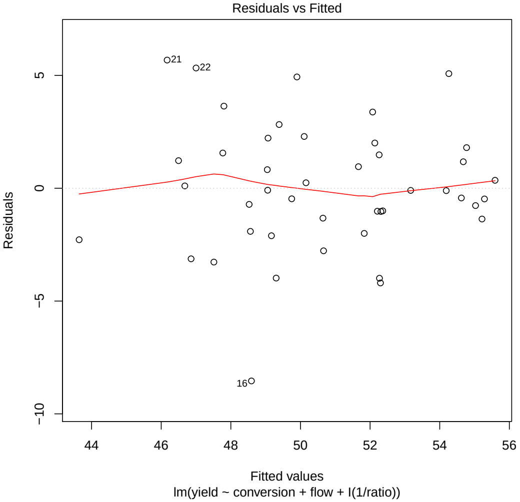
Q: Do the data support the idea that yield depend on the reciprocal of ratio?

```
> recip.LM = lm(yield~ conversion + flow +
+                     I(1/ratio), data = chem_pro.df)

> summary(recip.LM)
```

```
Call:
lm(formula = yield ~ conversion + flow + I(1/ratio), data = chem_pro.df)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5413 -1.5016 -0.1017  1.6203  5.6798

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept) 105.935729  10.694444   9.906 2.54e-12 ***
conversion   -4.263058   0.961839  -4.432 7.08e-05 ***
flow         -0.011542   0.003921  -2.944  0.00537 **
I(1/ratio)   -0.345478   0.155987  -2.215  0.03253 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.943 on 40 degrees of freedom
Multiple R-squared:  0.519,     Adjusted R-squared:  0.4829
F-statistic: 14.39 on 3 and 40 DF,  p-value: 1.663e-06
```

Q: Can we trust this model?

```
> plot(recip.LM, which = 1)
```



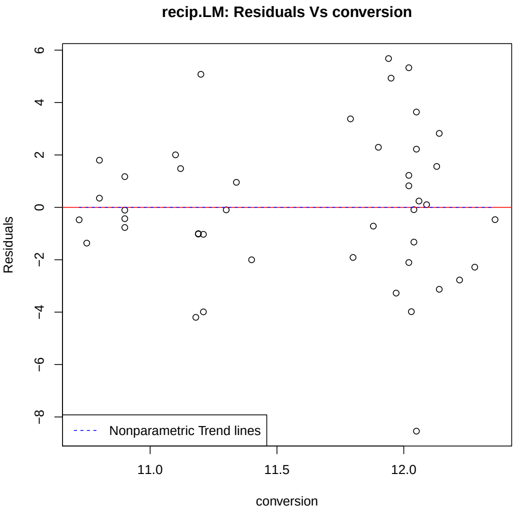
- It seems observation 16 is an outlier,

```
> recip_outlier_index = which(res < -6)
```

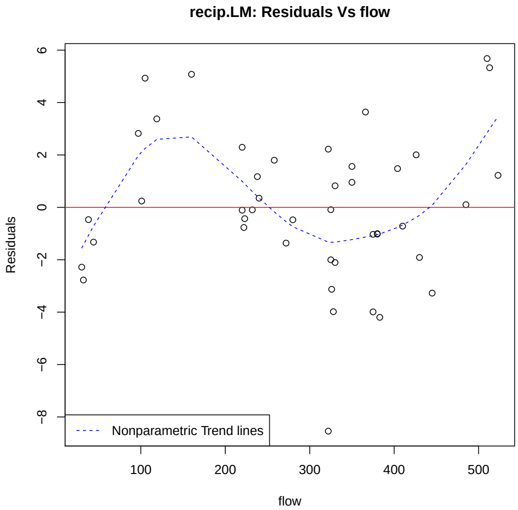
- We can try to improve the model by considering transformations on each of

  `conversion`, `flow` and `ratio`

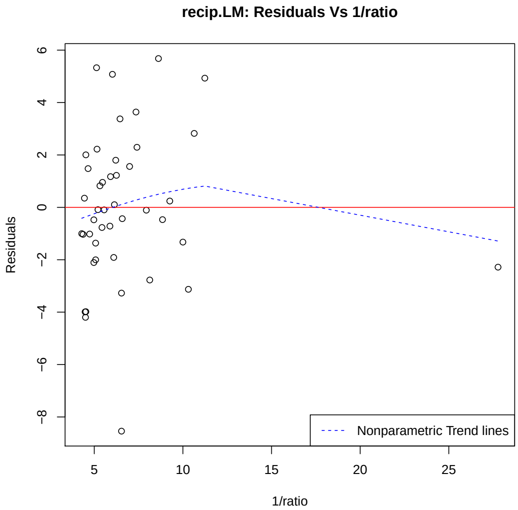
```
> res = recip.LM$residuals
>
> plot(chem_pro.df$conversion, res,
+      xlab = "conversion", ylab = "Residuals",
+      main = "recip.LM: Residuals Vs conversion")
> abline(h = 0, col = "red")

> lines(smooth.spline(chem_pro.df$conversion, res),
+       col = "blue", lty = 2)
> legend("bottomleft", lty = 2, col = 4,
+        c("Nonparametric Trend lines"))
```



- Notice the outlier and the possibility of having subgroups, but it seems no transformation is needed for `conversion`.



- It is a different story for `flow`, there seems to be a clear and strong cubic relationship.



- There seems to be a amount of curvature left, but it could be due to the high leverage point.

- Identifying the high leverage point,

```
> recip_hl_index = which(1/chem_pro.df$ratio > 25)
> recip_hl_index
```

```
[1] 6
```

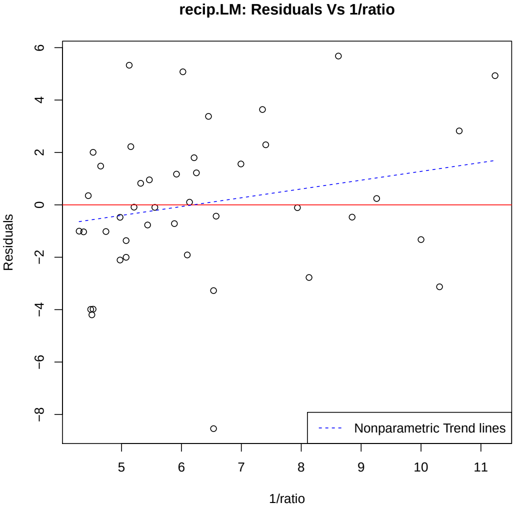
```
> ratio_unusual
```

```
[1] 6
```

```
> chem_pro.df[ratio_unusual,]
```

```
  yield conversion flow ratio
6 41.36      12.28   30 0.036
```

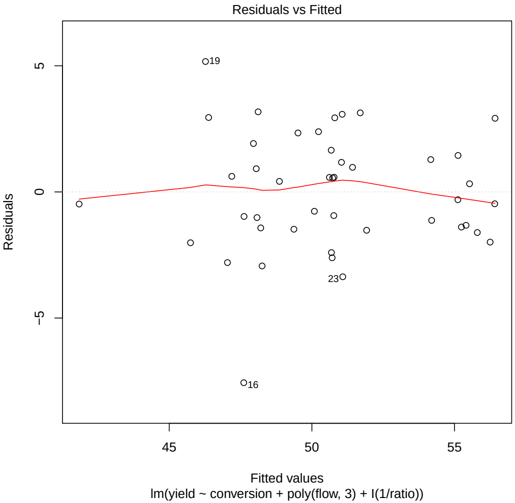
- We can see the effect this point on the trend line by leaving it out.

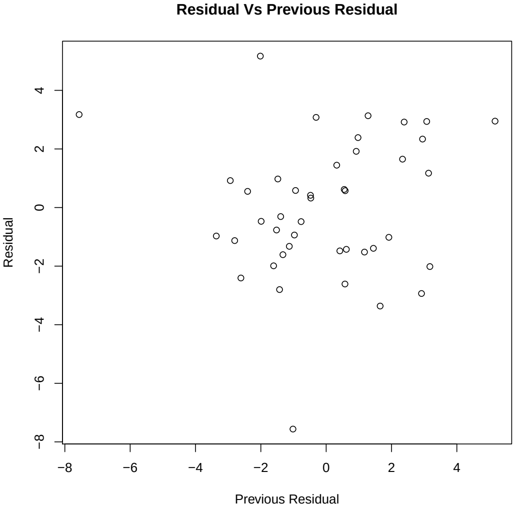


- It seems the small amount of curvature is caused by the 6th observation.
- No more transformation is need for `ratio`.

- Applying a cubic transformation on `flow`,

```
> cubic.LM = lm(yield~ conversion + poly(flow,3)
+                 + I(1/ratio), data = chem_pro.df)
```

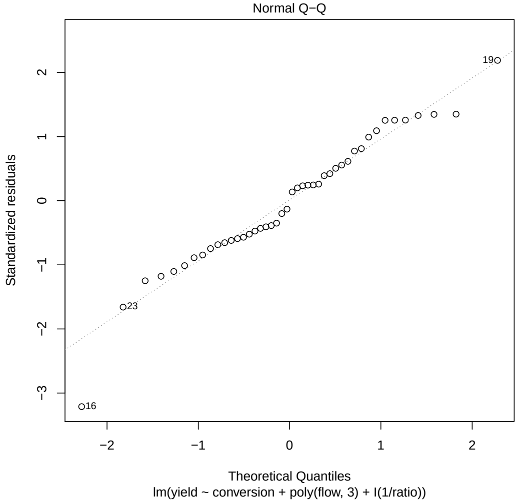




- Other than the outlier, Residual Vs Previous Residual plot seems to suggest no correlation.

- Other than the outlier, QQ-normal plot seems to indicate normality.

```
> plot(cubic.LM, which = 2)
```



```
> summary(cubic.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio),
    data = chem_pro.df)

Residuals:
   Min     1Q Median     3Q    Max
-7.562 -1.440  0.007  1.497  5.170

Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)    101.6345    10.4408   9.734 7.18e-12 ***
conversion      -4.1717     0.9105  -4.582 4.86e-05 ***
poly(flow, 3)1 -10.4209     3.0096  -3.462 0.001340 **
poly(flow, 3)2   3.7437     3.1710   1.181 0.245090
poly(flow, 3)3  10.0417     2.5762   3.898 0.000382 ***
I(1/ratio)      -0.3644     0.1404  -2.594 0.013386 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.469 on 38 degrees of freedom
Multiple R-squared:  0.6783,    Adjusted R-squared:  0.6359
F-statistic: 16.02 on 5 and 38 DF,  p-value: 1.742e-08
```

- This model is reasonably good, its explains 67.83% of variability in `yield`.
- The data support the claim that `yield` relates to `1/ratio` under this model.

- We press on and exam the other claim, that is, yield is related to

conversion*flow

```
> prod.LM =
+   lm(yield~ conversion + poly(flow,3) +
+         I(1/ratio) + I(flow*conversion),
+       data = chem_pro.df)
```

- After examining the residuals, it seems all assumptions are satisfied, but

```
> summary(prod.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio) +
    I(flow * conversion), data = chem_pro.df)

Coefficients:
                      Estimate Std. Error t value Pr(>t)
(Intercept)           99.42924   10.67628   9.313 3.08e-11 ***
conversion            -0.18663    4.11546  -0.045  0.96407
poly(flow, 3)1       127.62434  139.06123   0.918  0.36469
poly(flow, 3)2         1.73942    3.75947   0.463  0.64631
poly(flow, 3)3        13.30024    4.17243   3.188  0.00291 **
I(1/ratio)            -0.35397    0.14086  -2.513  0.01646 *
I(flow * conversion)  -0.01305    0.01314  -0.993  0.32720
```

- Notice conversion and flow were highly significant

```
> conversion = chem_pro.df$conversion
>
> flow = poly(chem_pro.df$flow,3)
>
> recip_ratio = 1/chem_pro.df$ratio
>
> prod = chem_pro.df$flow * chem_pro.df$conversion
>
> X = cbind(conversion, flow, recip_ratio, prod)
```

- Recall variance inflation factor of more than 10 is considered to be large,

```
> VIF = diag(solve(cor(X)))
>
> VIF
```

```
conversion           1           2           3 recip_ratio        prod
 32.276768 3169.973656    2.316849    2.853786    1.932279 3132.705871
```

- Eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ confirms, we definitely have collinearity problem

```
> eigen.stuff = eigen(cor(X))
>
> eigen.stuff$values
```

```
[1] 2.4837616083 1.7021510928 1.0013419291 0.4480934145 0.3644941045 0.0001578508
```

```
> round(eigen.stuff$vectors, 2)
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,]  0.28 -0.56  0.03  0.76  0.16  0.07
[2,] -0.55 -0.35  0.00 -0.17  0.20  0.71
[3,]  0.23 -0.57 -0.35 -0.36 -0.61 -0.01
[4,] -0.09  0.21 -0.94  0.18  0.20  0.02
[5,]  0.52 -0.19 -0.04 -0.48  0.68  0.00
[6,] -0.53 -0.40 -0.02 -0.08  0.24 -0.70
```

- It seems we have to go without `conversion*flow`.

- It seems that `conversion*flow` is unnecessary according to this dataset.
- We move back to the cubic model,

```
> cubic.LM = lm(yield~ conversion + poly(flow,3)
+                   + I(1/ratio), data = chem_pro.df)
```

- And investigate another aspect of model diagnostics for `cubic.LM`.
- High leverage points

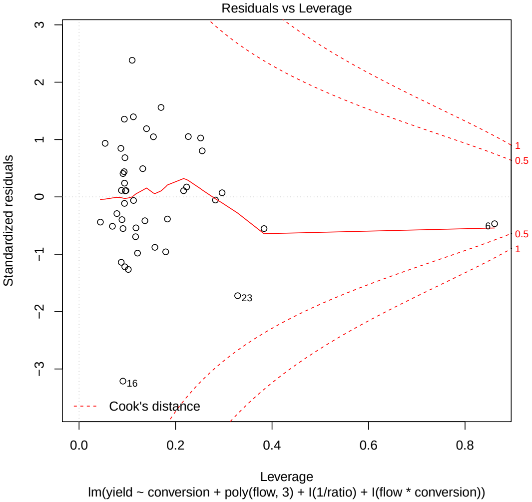
```
> plot(prod.LM, which = 5)
```

- Influential points

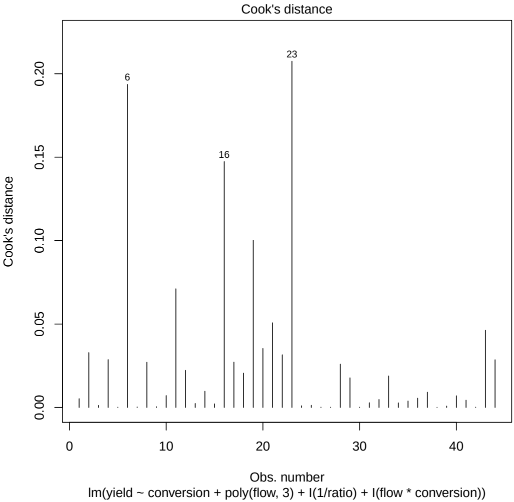
```
> plot(prod.LM, which = 4)
```

```
> influence.measures(cubic.LM)
```

- Recall high leverage points and influential points are different.



- Observations, 6, 16 and 23 are influential.



- The following

```
> influence.measures(cubic.LM)
```

gives a table with the followings as its columns

```
lm(formula = yield ~ conversion +
    poly(flow, 3) + I(1/ratio), data = chem_pro.df)

        dfb.1_
        dfb.cnvr
        dfb.p..3.1
        dfb.p..3.2
        dfb.p..3.3
        dfb.I.1.
        dffit
        cov.r
        cook.d
        hat
        inf
```

- Rows with `inf` = * is shown below

```
6   -0.184018  0.27195   -0.26389   4.31e-02    0.21870 -1.07461 -1.2852 8.077 0.280687
7   -0.009520  0.01655   -0.09130   6.73e-02   -0.05045 -0.06502  0.1246 1.630 0.002654
8   -0.029285 -0.00373    0.40018  -3.36e-01    0.26729  0.30531 -0.5513 1.667 0.051257
16   0.833470 -0.83617   -0.23336   8.30e-01   -0.01424 -0.08045 -1.1740 0.193 0.171884
```

- Observation 6 is high leverage influential point, and 16 is a influential outlier,

```
> recip_outlier_index
```

```
16
```

- Refit the model without the two observations,

```
> cubic.rm.LM =
+   lm(yield~conversion
+                    + poly(flow,3) + I(1/ratio),
+                    data = chem_pro.df[-c(6,16),])
>
> summary(cubic.rm.LM)
```

- The current cubic model and the previous model are shown below

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio),
    data = chem_pro.df[-c(6, 16), ])

Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)     96.0543     9.8763   9.726  1.3e-11 ***
conversion      -3.7628     0.9155  -4.110 0.000218 ***
poly(flow, 3)1  -9.2321     2.7876  -3.312 0.002117 **
poly(flow, 3)2   2.0092     2.6767   0.751 0.457747
poly(flow, 3)3   9.0940     2.3355   3.894 0.000410 ***
I(1/ratio)      -0.2025     0.2828  -0.716 0.478691
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.155 on 36 degrees of freedom
Multiple R-squared:  0.6727,    Adjusted R-squared:  0.6272
F-statistic:  14.8 on 5 and 36 DF,  p-value: 6.765e-08
```

```
Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)    101.6345    10.4408   9.734 7.18e-12 ***
conversion      -4.1717     0.9105  -4.582 4.86e-05 ***
poly(flow, 3)1 -10.4209     3.0096  -3.462 0.001340 **
poly(flow, 3)2   3.7437     3.1710   1.181 0.245090
poly(flow, 3)3  10.0417     2.5762   3.898 0.000382 ***
I(1/ratio)      -0.3644     0.1404  -2.594 0.013386 *
```

- It seems observation 6 and 16 are also problematic for prod.LM, however,

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio) +
    I(flow * conversion), data = chem_pro.df[-c(6, 16), ])

Coefficients:
                      Estimate Std. Error t value Pr(>t)
(Intercept)           94.57208    9.93354   9.520 3.02e-11 ***
conversion             0.15272    3.63512   0.042   0.9667
poly(flow, 3)1       118.75295  115.04943   1.032   0.3091
poly(flow, 3)2         0.42501    3.02399   0.141   0.8890
poly(flow, 3)3        12.19298    3.62966   3.359   0.0019 **
I(1/ratio)            -0.20802    0.28194  -0.738   0.4655
I(flow * conversion)  -0.01273    0.01144  -1.113   0.2734
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.148 on 35 degrees of freedom
Multiple R-squared:  0.6839,    Adjusted R-squared:  0.6297
F-statistic: 12.62 on 6 and 35 DF,  p-value: 1.567e-07
```

and eigenvalues/eigenvectors give similar conclusions.

```
> VIF = diag(solve(cor(X[-c(6,16),]))); VIF
```

```
 conversion           1           2           3 recip_ratio        prod
  31.416465 2889.153756    2.177048    2.895338    2.507123 2872.891795
```

- It seems 1/ratio is only significant because of those two points!

```
> chem_pro_final.LM =
+   lm(yield~conversion + poly(flow,3),
+      data = chem_pro.df[-c(6,16),])
> summary(chem_pro_final.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3), data = chem_pro.df[-c(6, 16), ])

Residuals:
    Min      1Q  Median      3Q     Max 
-4.1800 -1.4444  0.2131  1.7172  4.0394 

Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)     98.4299     9.2406  10.652 7.98e-13 ***
conversion      -4.0787     0.7968  -5.119 9.77e-06 ***
poly(flow, 3)1  -7.9682     2.1429  -3.718 0.000662 ***
poly(flow, 3)2   1.5336     2.5758   0.595 0.555213    
poly(flow, 3)3   8.5939     2.2138   3.882 0.000412 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.141 on 37 degrees of freedom
Multiple R-squared:  0.668,     Adjusted R-squared:  0.6322 
F-statistic: 18.61 on 4 and 37 DF,  p-value: 1.844e-08
```