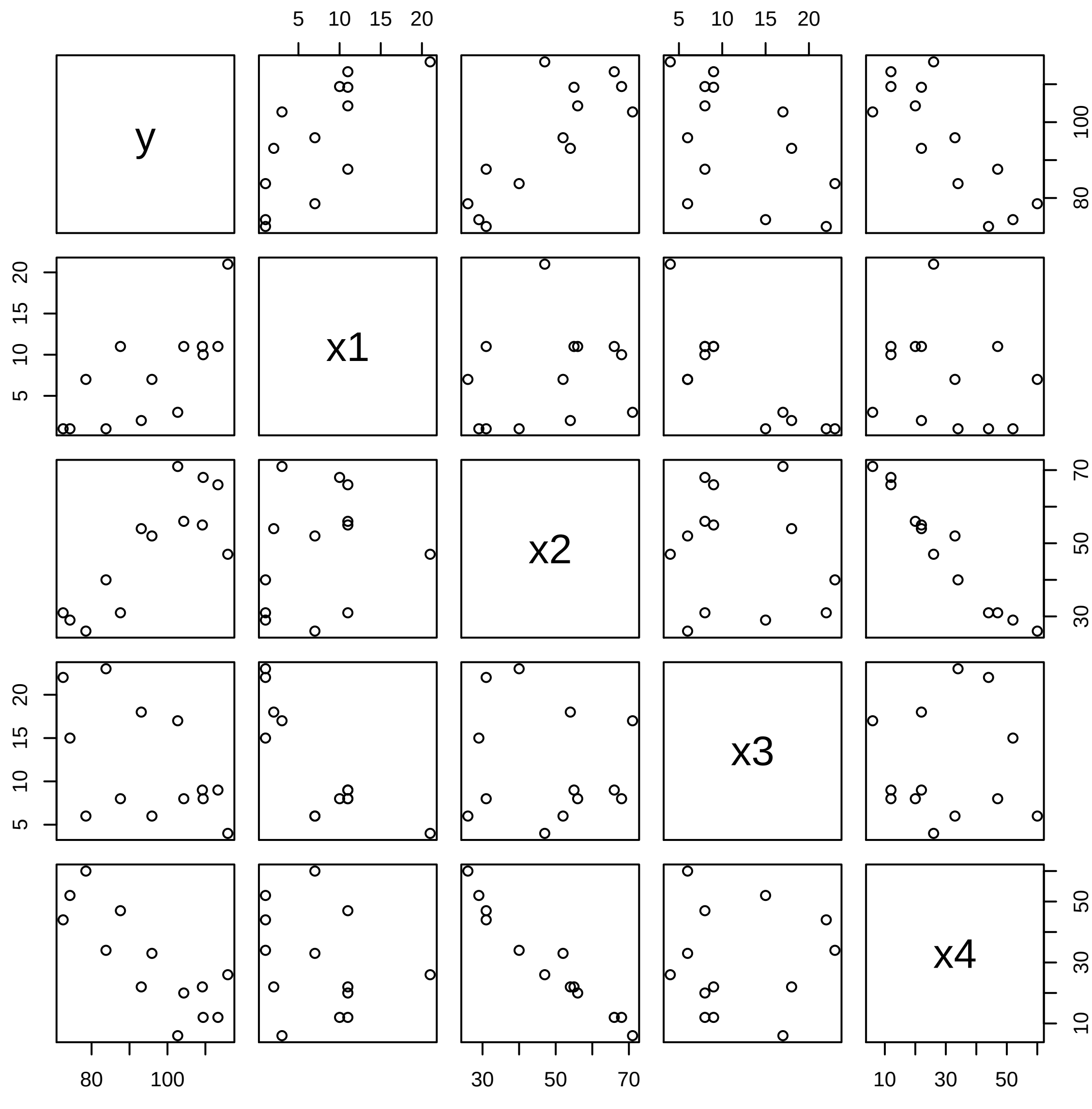
1 All possible regressions: The Hald Cement Data (MPV 9.1)

- Data concerning the heat evolved in calories per gram of cement (y) as a function of the amount of each of four ingredients in the mix
- x1=tricalcium aluminate, x2=tricalcium silicate, x3=tetracalcium alumino ferrite x4=dicalcium silicate




```
> dat=read.table("cement.dat", h=T); attach(dat)
> plot(dat)

> # create a function to obtain lm() information
```

```
> lm.info = function(g, sigma.full)
+ { # sigma: estimated sigma from the full model
+     p = ncol(g$model)
+     n = nrow(g$model)
+     if(n-p != g$df) stop("n-p != g$df")
+     rss =  sum(resid(g)^2)
+     gs= summary(g)
+     Cp = rss/sigma.full^2 - n + 2*p # Cp
+     aic = extractAIC(g,k=2)[2] # give AIC
+     bic = extractAIC(g,k=log(n))[2] # gives BIC
+     press = sum((resid(g)/(1-ls.diag(g)$hat))^2)  # press
+     c(p=p, df=n-p, ss=rss, rsq=gs$r.squared, rsq.a=gs$adj.r.squared, ms=rss/g$df,
+       Cp=Cp, aic=aic, bic=bic, press=press)
+ }
> # do all possible subset regressions
> full=lm(y~., dat)
> sigma.full = summary(full)$sigma
> info = NULL
> g=lm(y~1, dat)
> info = rbind(info, lm.info(g, sigma.full))
> subsets = list(1,2,3,4, c(1,2), c(1,3), c(1,4), c(2,3), c(2,4), c(3,4),
+  c(1,2,3), c(1,2,4), c(1,3,4), c(2,3,4), 1:4)
> for(i in 1:length(subsets)){
+     sub = c(1,subsets[[i]]+1)
+     g=lm(y~., dat[,sub])
+     info = rbind(info, lm.info(g, sigma.full))
+     }
> round(info, 2)
      p df      ss  rsq rsq.a     ms     Cp   aic   bic   press
 [1,] 1 12 2715.76 0.00  0.00 226.31 442.92 71.44 72.01 3187.25
 [2,] 2 11 1265.69 0.53  0.49 115.06 202.55 63.52 64.65 1699.61
 [3,] 2 11  906.34 0.67  0.64  82.39 142.49 59.18 60.31 1202.09
 [4,] 2 11 1939.40 0.29  0.22 176.31 315.15 69.07 70.20 2616.36
 [5,] 2 11  883.87 0.67  0.64  80.35 138.73 58.85 59.98 1194.22
 [6,] 3 10   57.90 0.98  0.97   5.79   2.68 25.42 27.11   93.88
 [7,] 3 10 1227.07 0.55  0.46 122.71 198.09 65.12 66.81 2218.12
 [8,] 3 10   74.76 0.97  0.97   7.48   5.50 28.74 30.44  121.22
 [9,] 3 10  415.44 0.85  0.82  41.54  62.44 51.04 52.73  701.74
[10,] 3 10  868.88 0.68  0.62  86.89 138.23 60.63 62.32 1461.81
[11,] 3 10  175.74 0.94  0.92  17.57  22.37 39.85 41.55  294.01
[12,] 4  9   48.11 0.98  0.98   5.35   3.04 25.01 27.27   90.00
[13,] 4  9   47.97 0.98  0.98   5.33   3.02 24.97 27.23   85.35
[14,] 4  9   50.84 0.98  0.98   5.65   3.50 25.73 27.99   94.54
[15,] 4  9   73.81 0.97  0.96   8.20   7.34 30.58 32.84  146.85
[16,] 5  8   47.86 0.98  0.97   5.98   5.00 26.94 29.77  110.35
>
> # all subset regressions, a simple way but don't tell which model is the best
> library(leaps)
> b <- regsubsets(y~x1+x2+x3+x4, dat)
> (rs <- summary(b))
Subset selection object
Call: regsubsets.formula(y ~ x1 + x2 + x3 + x4, dat)
4 Variables  (and intercept)
   Forced in Forced out
```

```
x1     FALSE      FALSE
x2     FALSE      FALSE
x3     FALSE      FALSE
x4     FALSE      FALSE
1 subsets of each size up to 4
Selection Algorithm: exhaustive
         x1  x2  x3  x4
1  ( 1 ) " " " " " " "*"
2  ( 1 ) "*" "*" " " " "
3  ( 1 ) "*" "*" " " "*"
4  ( 1 ) "*" "*" "*" "*"

> # compare two best models
> g6=lm(y~x1+x2, dat); summary(g6)   # model 6 chosen by Cp and BIC
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
x1           1.46831    0.12130   12.11 2.69e-07 ***
x2           0.66225    0.04585   14.44 5.03e-08 ***

Residual standard error: 2.406 on 10 degrees of freedom
Multiple R-Squared: 0.9787,Adjusted R-squared: 0.9744
F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09

> g13=lm(y~x1+x2+x4, dat); summary(g13) # model 13 chosen by MS, AIC and PRESS
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  71.6483    14.1424   5.066 0.000675 ***
x1            1.4519     0.1170  12.410 5.78e-07 ***
x2            0.4161     0.1856   2.242 0.051687 .
x4           -0.2365     0.1733  -1.365 0.205395

Residual standard error: 2.309 on 9 degrees of freedom
Multiple R-Squared: 0.9823,Adjusted R-squared: 0.9764
F-statistic: 166.8 on 3 and 9 DF,  p-value: 3.323e-08

> # check residual plots, appear to be okay
> par(mfrow=c(2,2))
> plot(g6, 1:4)
> plot(g13, 1:4)
> # predicted values are similar
> round(fitted(g6),2)
     1      2      3      4      5      6      7      8      9     10     11     12     13
 80.07  73.25 105.81  89.26  97.29 105.15 104.00  74.58  91.28 114.54  80.54 112.44 112.29
> round(fitted(g13),2)
     1      2      3      4      5      6      7      8      9     10     11     12     13
 78.44  72.87 106.19  89.40  95.64 105.30 104.13  75.59  91.82 115.55  81.70 112.24 111.62

> # vif's are different
> library(car)
> vif(g6)
      x1       x2
1.055129 1.055129
> vif(g13)
       x1        x2        x4
 1.066330 18.780309 18.940077
```

2 Stepwise regressions: The Hald Cement Data (MPV 9.2-9.4)

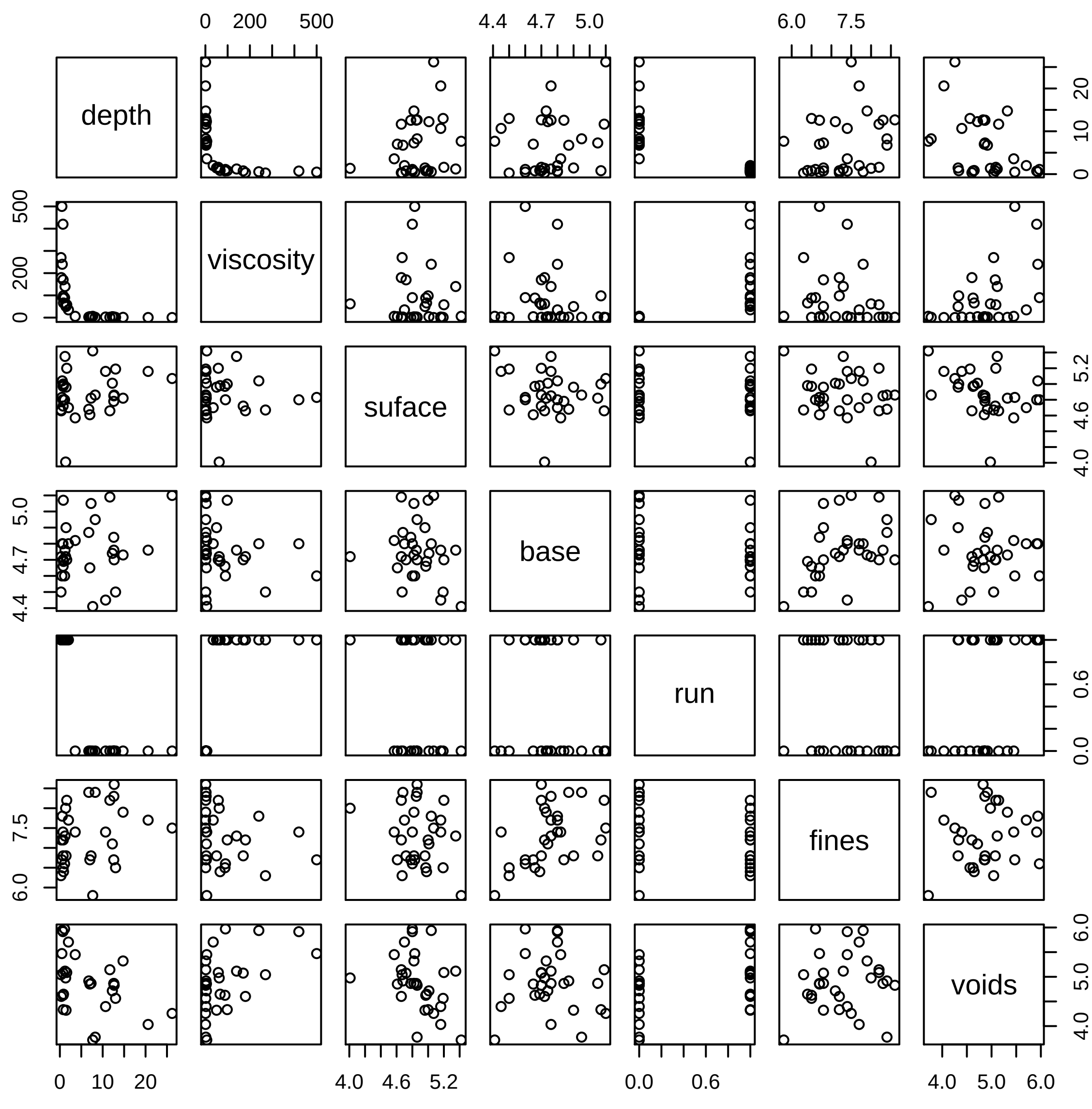
```
> ## forward selection using Cp
> g1=lm(y~1, dat)
> step(g1,   scope=~x1+x2+x3+x4, data=dat, direction="forward",  scale=sigma.full^2)  # Cp
Start:  AIC= 442.92
 y ~ 1

       Df Sum of Sq     RSS     Cp
+ x4    1   1831.90  883.87 138.73
+ x2    1   1809.43  906.34 142.49
+ x1    1   1450.08 1265.69 202.55
+ x3    1    776.36 1939.40 315.15
<none>              2715.76 442.92

Step:  AIC= 138.73
 y ~ x4

       Df Sum of Sq    RSS       Cp
+ x1    1    809.10  74.76   5.4959
+ x3    1    708.13 175.74  22.3731
+ x2    1     14.99 868.88 138.2259
<none>              883.87 138.7308

Step:  AIC= 5.5
 y ~ x4 + x1

       Df Sum of Sq    RSS     Cp
+ x2    1    26.789 47.973 3.0182
+ x3    1    23.926 50.836 3.4968
<none>              74.762 5.4959

Step:  AIC= 3.02
 y ~ x4 + x1 + x2

       Df Sum of Sq    RSS     Cp
<none>              47.973 3.0182
+ x3    1     0.109 47.864 5.0000

Call:
lm(formula = y ~ x4 + x1 + x2, data = dat)

Coefficients:
(Intercept)           x4           x1           x2
    71.6483      -0.2365       1.4519       0.4161

> ## backward elimination using Cp
> step(full,  data=dat, direction="backward", scale=sigma.full^2)  # Cp
Start:  AIC= 5
 y ~ x1 + x2 + x3 + x4

       Df Sum of Sq    RSS     Cp
- x3    1     0.109 47.973 3.0182
- x4    1     0.247 48.111 3.0413
```

```
- x2    1      2.972 50.836 3.4968
<none>               47.864 5.0000
- x1    1     25.951 73.815 7.3375

Step:  AIC= 3.02
 y ~ x1 + x2 + x4

       Df Sum of Sq    RSS       Cp
- x4    1      9.93  57.90   2.6782
<none>                47.97   3.0182
- x2    1     26.79  74.76   5.4959
- x1    1    820.91 868.88 138.2259

Step:  AIC= 2.68
 y ~ x1 + x2

       Df Sum of Sq     RSS       Cp
<none>                 57.90   2.6782
- x1    1    848.43  906.34 142.4864
- x2    1   1207.78 1265.69 202.5488

Call:
lm(formula = y ~ x1 + x2, data = dat)

Coefficients:
(Intercept)           x1           x2
    52.5773       1.4683       0.6623

> ## stepwise regression using Cp
> step(full, dat, direction="both",  scale=sigma.full^2)  # Cp
Start:  AIC= 5
 y ~ x1 + x2 + x3 + x4

       Df Sum of Sq    RSS     Cp
- x3    1      0.109 47.973 3.0182
- x4    1      0.247 48.111 3.0413
- x2    1      2.972 50.836 3.4968
<none>               47.864 5.0000
- x1    1     25.951 73.815 7.3375

Step:  AIC= 3.02
 y ~ x1 + x2 + x4

       Df Sum of Sq    RSS       Cp
- x4    1      9.93  57.90   2.6782
<none>                47.97   3.0182
- x2    1     26.79  74.76   5.4959
- x1    1    820.91 868.88 138.2259

Step:  AIC= 2.68
 y ~ x1 + x2

       Df Sum of Sq     RSS       Cp
<none>                 57.90   2.6782
```

```
- x1    1    848.43  906.34 142.4864
- x2    1   1207.78 1265.69 202.5488

Call:
lm(formula = y ~ x1 + x2, data = dat)

Coefficients:
(Intercept)           x1           x2
    52.5773       1.4683       0.6623

>
> ## stepwise regression using AIC or BIC
> step(g1,  scope=~x1+x2+x3+x4, data=dat, direction="both", trace=0)  # AIC

Call:
lm(formula = y ~ x4 + x1 + x2, data = dat)

Coefficients:
(Intercept)           x4           x1           x2
    71.6483      -0.2365       1.4519       0.4161

> step(g1,  scope=~x1+x2+x3+x4, data=dat, direction="both",  k=log(nrow(dat)), trace=0)  # BIC

Call:
lm(formula = y ~ x1 + x2, data = dat)

Coefficients:
(Intercept)           x1           x2
    52.5773       1.4683       0.6623
```

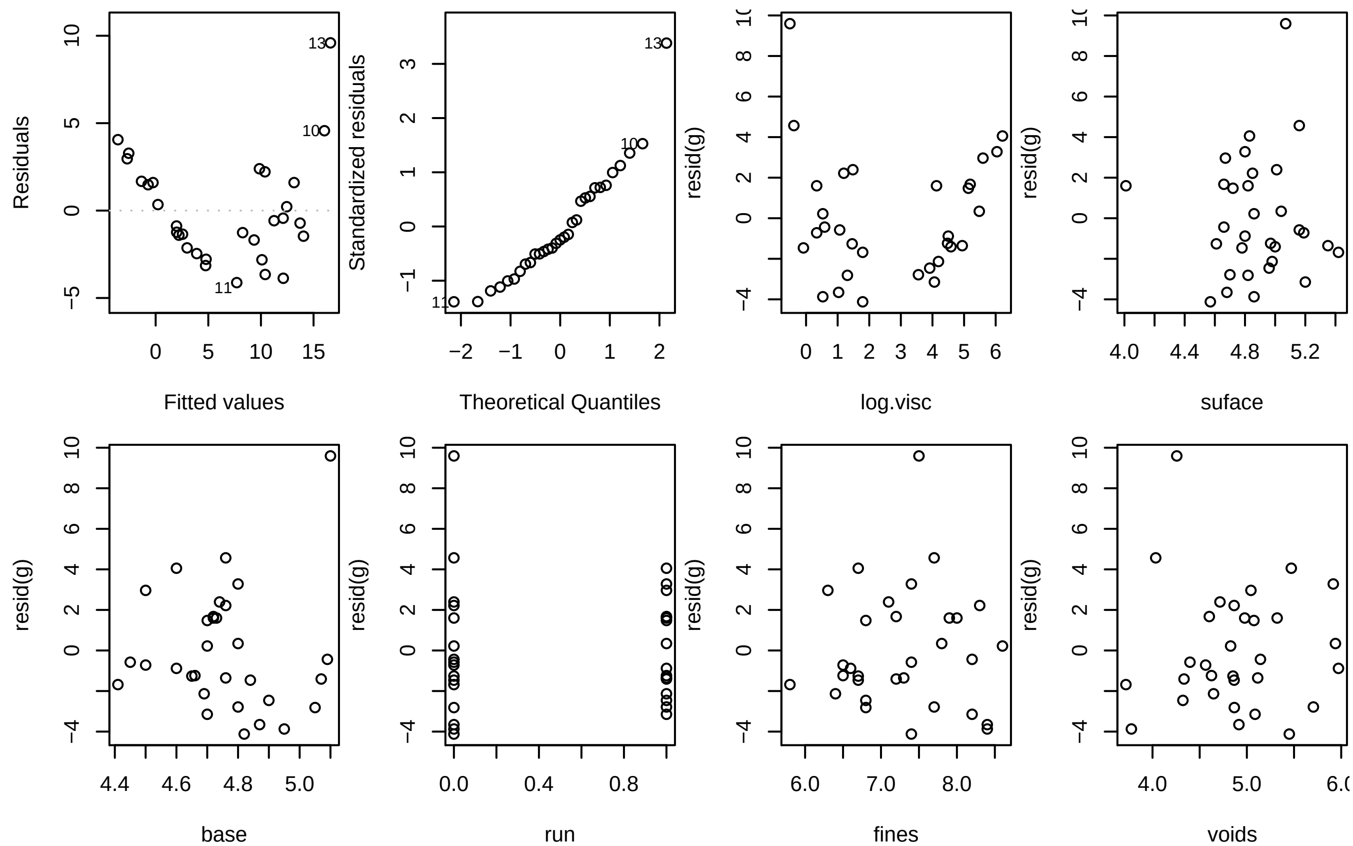
3 Case study: The asphalt data (MPV 9.4)

- Data concerning the rut depth of 31 asphalt pavements prepared under different conditions specified by 5 regressors.
- A 6th regressor is used as an indicator variable to separate the data into 2 sets of runs.
- y=rut depth per million wheel passes, x1=viscosity of the asphalt, x2=percentage of asphalt in the surface courses, x3=percentage of asphalt in the base course, x4=the run (0 or 1), x5=percentage of fines in the surface course, x6=percentage of voids in the surface course.

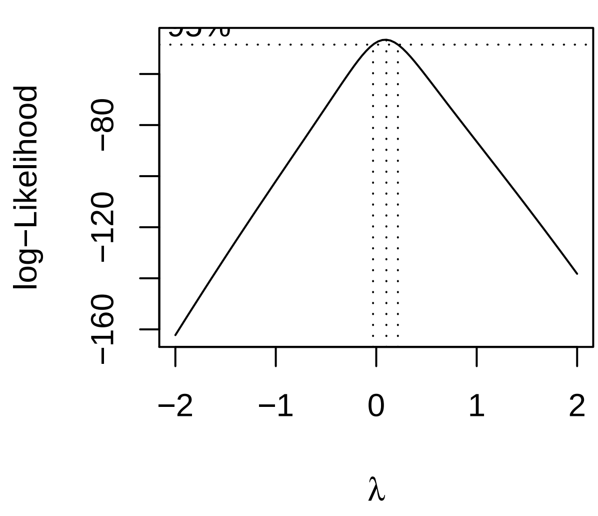



```
> dat=read.table("asphalt.dat", h=T); attach(dat)
> plot(dat)
```

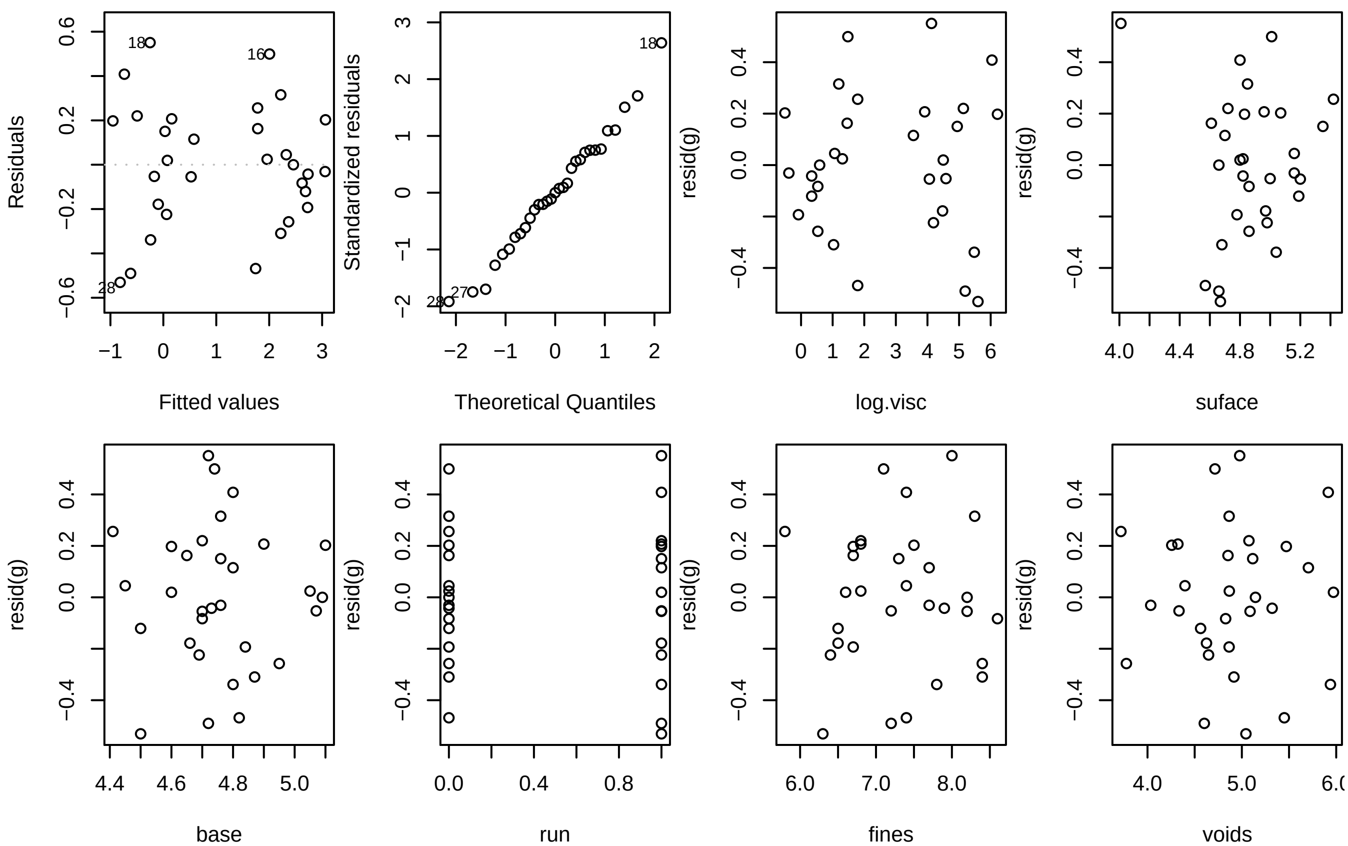
```
> # see if we need transform the predictors
> library(alr3)  # bctrans
> library(MASS)  # boxcox
> ans=bctrans(depth~viscosity+suface+base+run+fines+voids, dat)  #
Error in bctrans1(mf, Y = y, ..., call = match.call(expand.dots = TRUE)) :
All values must be > 0; use family="yeo.johnson"
> ans=bctrans(depth~viscosity+suface+base+fines+voids, dat) #
> summary(ans)
box.cox Transformations to Multinormality

          Est.Power Std.Err. Wald(Power=0) Wald(Power=1)
viscosity   -0.0224   0.1063       -0.2105       -9.6205
suface       4.7640   2.3060        2.0659        1.6323
base        -3.5401   4.2424       -0.8344       -1.0702
fines       -0.7789   1.8330       -0.4250       -0.9705
voids        0.5065   1.2129        0.4176       -0.4068
                                  LRT df   p.value
LR test, all lambda equal 0  5.860003  5 0.3200805
LR test, all lambda equal 1 89.039393  5 0.0000000

> lrt.bctrans(ans,lrt=list(c(0,1,1,1,1)))
                                  LRT df   p.value
LR test, all lambda equal 0  5.860003  5 0.3200805
LR test, all lambda equal 1 89.039393  5 0.0000000
LR test, lambda = 0 1 1 1 1  4.842049  5 0.4354595

> # log transform for viscosity
> log.visc = log(viscosity)
> g=lm(depth~log.visc+suface+base+run+fines+voids)
> summary(g)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -14.9592    25.2881  -0.592  0.55968
log.visc     -3.1515     0.9194  -3.428  0.00220 **
suface        3.9706     2.4966   1.590  0.12484
base          1.2631     3.9703   0.318  0.75312
run           1.9655     3.6472   0.539  0.59492
fines         0.1164     1.0124   0.115  0.90939
voids         0.5893     1.3244   0.445  0.66036

Residual standard error: 3.324 on 24 degrees of freedom
Multiple R-Squared: 0.806,Adjusted R-squared: 0.7575
F-statistic: 16.62 on 6 and 24 DF,  p-value: 1.743e-07

> # check residual plots to see if need transform the response
> par(mfrow=c(2,4)); par(mar=c(5,4,0,0)+.1)
> plot(g, 2:1)
> plot(log.visc, resid(g))
> plot(suface, resid(g))
> plot(base, resid(g))
> plot(run, resid(g))
> plot(fines, resid(g))
> plot(voids, resid(g))
```




```
> # transform the response using boxcox method
> boxcox(g)
```



```
> ## now refit with transformed response
> g=lm(log(depth)~log.visc+suface+base+run+fines+voids)
> summary(g)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.23294    2.34970  -0.525    0.605
log.visc    -0.55769    0.08543  -6.528 9.45e-07 ***
suface       0.58358    0.23198   2.516    0.019 *
base        -0.10337    0.36891  -0.280    0.782
run         -0.34005    0.33889  -1.003    0.326
fines        0.09775    0.09407   1.039    0.309
voids        0.19885    0.12306   1.616    0.119

Residual standard error: 0.3088 on 24 degrees of freedom
Multiple R-Squared: 0.961,Adjusted R-squared: 0.9512
F-statistic: 98.47 on 6 and 24 DF,  p-value: 1.059e-15

> par(mfrow=c(2,4)); par(mar=c(5,4,0,0)+.1)
> plot(g, 1:2)
> plot(log.visc, resid(g))
> plot(suface, resid(g))
> plot(base, resid(g))
> plot(run, resid(g))
> plot(fines, resid(g))
> plot(voids, resid(g))
```



```
> # do model/variable selection
> library(leaps)
> b <- regsubsets(log(depth)~log.visc+suface+base+run+fines+voids, dat)
> summary(b)
Subset selection object
Call: regsubsets.formula(log(depth) ~ log.visc + suface + base + run +
    fines + voids, dat)
6 Variables  (and intercept)
         Forced in Forced out
log.visc     FALSE      FALSE
suface       FALSE      FALSE
base         FALSE      FALSE
run          FALSE      FALSE
fines        FALSE      FALSE
voids        FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: exhaustive
         log.visc suface base run fines voids
1  ( 1 ) "*"      " "    " "  " " " "   " "
2  ( 1 ) "*"      "*"    " "  " " " "   " "
3  ( 1 ) "*"      "*"    " "  " " " "   "*"
4  ( 1 ) "*"      "*"    " "  "*" " "   "*"
5  ( 1 ) "*"      "*"    " "  "*" "*"   "*"
6  ( 1 ) "*"      "*"    "*"  "*" "*"   "*"
>
> # stepwise regressions using AIC, BIC and Cp
> full=lm(log(depth)~log.visc+suface+base+run+fines+voids)
> g1=lm(log(depth)~1, dat)
> step(g1, scope=~log.visc+suface+base+run+fines+voids, data=dat, direction="both", trace=0)  # AIC

Call:
lm(formula = log(depth) ~ log.visc + suface + voids, data = dat)

Coefficients:
(Intercept)     log.visc       suface        voids
    -1.0208      -0.6465       0.5555       0.2448

> step(full, dat, direction="both",  k=log(nrow(dat)), trace=0)  # BIC

Call:
lm(formula = log(depth) ~ log.visc + suface + voids)

Coefficients:
(Intercept)     log.visc       suface        voids
    -1.0208      -0.6465       0.5555       0.2448

> step(full, dat, direction="both", scale=sigma.hat(full)^2, trace=0)  # Cp

Call:
lm(formula = log(depth) ~ log.visc + suface + voids)

Coefficients:
(Intercept)     log.visc       suface        voids
    -1.0208      -0.6465       0.5555       0.2448
```

```
> ## all methods choose the same model
> g=lm(log(depth)~log.visc+suface+voids)
> summary(g)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.02079    1.36430  -0.748   0.4608
log.visc    -0.64649    0.02879 -22.458   <2e-16 ***
suface       0.55547    0.22044   2.520   0.0180 *
voids        0.24479    0.11560   2.118   0.0436 *

Residual standard error: 0.3025 on 27 degrees of freedom
Multiple R-Squared: 0.9579,Adjusted R-squared: 0.9532
F-statistic: 204.6 on 3 and 27 DF,  p-value: < 2.2e-16

> round(lm.info(g, sigma.hat(full)),4)
       p       df       ss      rsq    rsq.a       ms       Cp      aic      bic    press
  4.0000  27.0000   2.4706   0.9579   0.9532   0.0915   2.9066 -70.4155 -64.6795   3.7515

> # check residual plots again for the final model
> par(mfrow=c(2,3))
> plot(g, 1:2)
> plot(log.visc, resid(g))
> plot(suface, resid(g))
> plot(voids, resid(g))
```