```
set.seed(22)
n <- 100
##################### Example 1 ######################
##### Variable Selection on simulated data #########
#####################################################
X <- as.data.frame(matrix(runif(10*n), nrow=n)) ##matrix 100*10
X[,4] <- X[,1]+rnorm(n,0,.1)
names(X) <- paste("x", 1:10, sep="")
> X
          x1          x2          x3          x4         x5         x6
1  0.30427684 0.15958112 0.043411845  0.4068466618 0.761921449 0.1680194580
2  0.47473891 0.14495172 0.429162335  0.2983675709 0.884934078 0.5247247852
3  0.99352577 0.70091386 0.212081804  0.9778346966 0.020966388 0.4264585804
          x7         x8          x9        x10
1  0.659121157 0.601975398 3.482328e-01 0.23588319
2  0.398862305 0.110264649 5.695797e-01 0.41295338
3  0.813279488 0.852846867 9.720615e-01 0.23320095

> true <- 10 + 2*X[,1] + X[,2] + 3*X[,3]
> y <- true + rnorm(n, 0, 1.1)
> ex.data <- as.data.frame(cbind(X,y))
> ex.data[1:3,]
       x1       x2        x3       x4        x5       x6       x7
1 0.3042768 0.1595811 0.04341185 0.4068467 0.76192145 0.1680195 0.6591212
2 0.4747389 0.1449517 0.42916233 0.2983676 0.88493408 0.5247248 0.3988623
3 0.9935258 0.7009139 0.21208180 0.9778347 0.02096639 0.4264586 0.8132795
       x8       x9      x10       y
1 0.6019754 0.3482328 0.2358832 12.75293
2 0.1102646 0.5695797 0.4129534 12.06790
3 0.8528469 0.9720615 0.2332010 13.97815
>

## Notice that the x5-x10 predictors are completely uninformative
## (i.e., they are not included in the "true" model for y)

> ### Fit full model ###
> ans.all <- lm(y ~X[,1]+X[,2]+X[,3]+X[,4]+X[,5]+X[,6]+X[,7]+X[,8]+X[,9]+X[,10])
> summary(ans.all)

Call:
lm(formula = y ~ X[, 1] + X[, 2] + X[, 3] + X[, 4] + X[, 5] +
    X[, 6] + X[, 7] + X[, 8] + X[, 9] + X[, 10])

Residuals:
    Min      1Q  Median      3Q     Max
-2.1835 -0.7360  0.1041  0.6419  3.0108

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4156     0.6047  17.224  < 2e-16 ***
```

```
X[, 1]       2.0145     1.0671   1.888   0.0623 .
X[, 2]       0.8311     0.3982   2.087   0.0398 *
X[, 3]       3.1304     0.3945   7.935  5.9e-12 ***
X[, 4]      -0.4088     1.0204  -0.401   0.6896
X[, 5]      -0.2012     0.3837  -0.524   0.6014
X[, 6]       0.1129     0.3944   0.286   0.7754
X[, 7]       0.6621     0.4094   1.617   0.1094
X[, 8]      -0.2272     0.4384  -0.518   0.6057
X[, 9]      -0.3722     0.4202  -0.886   0.3782
X[, 10]      -0.4159     0.4028  -1.033   0.3046
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.09 on 89 degrees of freedom
Multiple R-squared:  0.519,     Adjusted R-squared:  0.4649
F-statistic: 9.602 on 10 and 89 DF,  p-value: 1.149e-10


> ## calculate mse from true line
> mse.all <- sum((ans.all$fitted - true)^2)/n
> mse.all
[1] 0.09172456
>
### Perform Best Subsets search ###

## Install package "leaps"  (only need to do this the first time)
#install.packages("leaps")

## Load library "leaps"
library(leaps)

                # can use method = "Cp", "adjr2", or "r2"
ans.bs <- leaps(X, y, method="Cp", nbest=5)

> ## Get the 5 "best" models according to Cp
> ans.bs$which[order(ans.bs$Cp)[1:5],]
    1     2     3     4     5     6     7     8     9     A
3 TRUE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
4 TRUE  TRUE TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
4 TRUE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
5 TRUE  TRUE TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
2 TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> ans.bs$Cp[order(ans.bs$Cp)[1:5]]
[1] 2.088167 2.130498 2.525953 2.645679 2.897483

> ## Get the 5 "best" models according to adjusted R^2
> ans.r2 <- leaps(X, y, method="adjr2", nbest=5)
> ans.r2$which[order(-ans.r2$adjr2)[1:5],]
    1     2     3     4     5     6     7     8     9     A
5 TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  TRUE
```

```
6 TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE  TRUE  TRUE
4 TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
5 TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE  TRUE FALSE
6 TRUE TRUE TRUE FALSE FALSE FALSE TRUE  TRUE FALSE  TRUE
> ans.r2$adjr2[order(-ans.r2$adjr2)[1:5]]
[1] 0.4840300 0.4830352 0.4810984 0.4802560 0.4801410


> ### Perform Stepwise search ###
>
> ## Must also specify the "full" model
> upper <- formula(~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10)
> lower <- formula(~ 1)
> ## Must first create a "base" model for step function
> ## This one will start with the "null" model.  (e.g. Forward Stagewise)
> ans0 <- lm(y ~ 1, data=ex.data)
>
> ##Now conduct stepwise search
>                 ## Can use direction = "both", "backward", or "forward"
> ans.sw <- step(ans0, scope=list(lower=lower, upper=upper), direction = "both")
Start:  AIC=80.85
y ~ 1

      Df Sum of Sq    RSS    AIC
+ x3   1    82.215 137.81 36.067
+ x1   1    17.569 202.45 74.532
+ x4   1    11.505 208.51 77.484
+ x2   1    10.977 209.04 77.737
<none>             220.02 80.855
+ x7   1     3.765 216.25 81.129
+ x9   1     3.205 216.81 81.387
+ x6   1     0.220 219.80 82.755
+ x5   1     0.188 219.83 82.769
+ x8   1     0.171 219.85 82.777
+ x10  1     0.053 219.97 82.831

Step:  AIC=36.07
y ~ x3

      Df Sum of Sq    RSS    AIC
+ x1   1    22.581 115.22 20.171
+ x4   1    17.697 120.11 24.322
+ x2   1     5.970 131.84 33.638
<none>             137.81 36.067
+ x7   1     2.210 135.59 36.450
+ x9   1     0.443 137.36 37.745
+ x6   1     0.357 137.45 37.807
+ x5   1     0.201 137.60 37.921
+ x10  1     0.113 137.69 37.985
+ x8   1     0.050 137.75 38.030
```

```
- x3    1    82.215 220.02 80.855

Step:  AIC=20.17
y ~ x3 + x1

       Df Sum of Sq    RSS    AIC
+ x2    1     3.341 111.88 19.229
<none>              115.22 20.171
+ x7    1     1.326 113.90 21.013
+ x10   1     1.095 114.13 21.216
+ x5    1     0.529 114.69 21.711
+ x9    1     0.404 114.82 21.820
+ x4    1     0.290 114.94 21.919
+ x6    1     0.211 115.01 21.987
+ x8    1     0.140 115.08 22.050
- x1    1    22.581 137.81 36.067
- x3    1    87.226 202.45 74.532

Step:  AIC=19.23
y ~ x3 + x1 + x2

       Df Sum of Sq    RSS    AIC
+ x7    1     2.328 109.56 19.126
<none>              111.88 19.229
+ x10   1     1.858 110.03 19.555
- x2    1     3.341 115.22 20.171
+ x5    1     0.398 111.49 20.873
+ x6    1     0.341 111.54 20.924
+ x9    1     0.314 111.57 20.948
+ x4    1     0.310 111.57 20.951
+ x8    1     0.222 111.66 21.030
- x1    1    19.952 131.84 33.638
- x3    1    82.688 194.57 72.563

Step:  AIC=19.13
y ~ x3 + x1 + x2 + x7

       Df Sum of Sq    RSS    AIC
<none>              109.56 19.126
- x7    1     2.328 111.88 19.229
+ x10   1     1.766 107.79 19.502
+ x9    1     0.977 108.58 20.230
+ x8    1     0.413 109.14 20.748
+ x5    1     0.403 109.15 20.758
+ x4    1     0.364 109.19 20.793
+ x6    1     0.345 109.21 20.811
- x2    1     4.342 113.90 21.013
- x1    1    18.467 128.02 32.704
- x3    1    80.138 189.69 72.024
> summary(ans.sw)
```

```
Call:
lm(formula = y ~ x3 + x1 + x2 + x7, data = ex.data)

Residuals:
   Min      1Q  Median      3Q     Max
-2.0901 -0.7932  0.1078  0.6453  3.1608

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.9728     0.3656  27.282  < 2e-16 ***
x3            3.1782     0.3813   8.336 5.87e-13 ***
x1            1.5363     0.3839   4.002 0.000125 ***
x2            0.7475     0.3852   1.940 0.055291 .
x7            0.5479     0.3856   1.421 0.158652
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.074 on 95 degrees of freedom
Multiple R-squared:  0.5021,    Adjusted R-squared:  0.4811
F-statistic: 23.95 on 4 and 95 DF,  p-value: 1.026e-13

>

> ## Now start with the "full" model.  (e.g. Backward Stagewise)
> ans1 <- lm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10, data=ex.data)
> ans.sw1 <- step(ans1, scope=list(lower=lower, upper=upper), direction =
"backward")
Start:  AIC=27.67
y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

       Df Sum of Sq    RSS    AIC
- x6    1     0.097 105.93 25.761
- x4    1     0.191 106.02 25.850
- x8    1     0.319 106.15 25.971
- x5    1     0.327 106.16 25.978
- x9    1     0.933 106.77 26.547
- x10   1     1.268 107.10 26.860
<none>              105.83 27.669
- x7    1     3.110 108.94 28.566
- x1    1     4.238 110.07 29.596
- x2    1     5.179 111.01 30.447
- x3    1    74.875 180.71 79.171

Step:  AIC=25.76
y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8 + x9 + x10

       Df Sum of Sq    RSS    AIC
- x4    1     0.155 106.08 23.907
- x5    1     0.371 106.30 24.111
```

```
- x8    1     0.484 106.42 24.218
- x9    1     1.009 106.94 24.709
- x10   1     1.276 107.21 24.959
<none>              105.93 25.761
- x7    1     3.182 109.11 26.721
- x1    1     4.162 110.09 27.616
- x2    1     5.120 111.05 28.482
- x3    1    74.922 180.85 77.251

Step:  AIC=23.91
y ~ x1 + x2 + x3 + x5 + x7 + x8 + x9 + x10

      Df Sum of Sq    RSS    AIC
- x5    1     0.350 106.44 22.237
- x8    1     0.555 106.64 22.429
- x9    1     1.037 107.12 22.880
- x10   1     1.386 107.47 23.205
<none>              106.08 23.907
- x7    1     3.167 109.25 24.849
- x2    1     5.142 111.23 26.640
- x1    1    19.626 125.71 38.882
- x3    1    75.856 181.94 75.851

Step:  AIC=22.24
y ~ x1 + x2 + x3 + x7 + x8 + x9 + x10

      Df Sum of Sq    RSS    AIC
- x8    1     0.413 106.85 20.625
- x9    1     1.012 107.45 21.183
- x10   1     1.652 108.09 21.777
<none>              106.44 22.237
- x7    1     3.102 109.54 23.109
- x2    1     5.333 111.77 25.126
- x1    1    19.420 125.86 36.996
- x3    1    75.570 182.01 73.887

Step:  AIC=20.62
y ~ x1 + x2 + x3 + x7 + x9 + x10

      Df Sum of Sq    RSS    AIC
- x9    1     0.941 107.79 19.502
- x10   1     1.729 108.58 20.230
<none>              106.85 20.625
- x7    1     2.874 109.72 21.279
- x2    1     5.162 112.01 23.342
- x1    1    19.385 126.23 35.297
- x3    1    76.161 183.01 72.437

Step:  AIC=19.5
y ~ x1 + x2 + x3 + x7 + x10
```

```
      Df Sum of Sq    RSS    AIC
- x10  1     1.766 109.56 19.126
<none>             107.79 19.502
- x7   1     2.236 110.03 19.555
- x2   1     5.134 112.92 22.154
- x1   1    19.643 127.43 34.242
- x3   1    80.261 188.05 73.154

Step:  AIC=19.13
y ~ x1 + x2 + x3 + x7

      Df Sum of Sq    RSS    AIC
<none>             109.56 19.126
- x7   1     2.328 111.88 19.229
- x2   1     4.342 113.90 21.013
- x1   1    18.467 128.02 32.704
- x3   1    80.138 189.69 72.024



> summary(ans.sw1)

Call:
lm(formula = y ~ x1 + x2 + x3 + x7, data = ex.data)

Residuals:
   Min      1Q  Median      3Q     Max
-2.0901 -0.7932  0.1078  0.6453  3.1608

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.9728     0.3656  27.282  < 2e-16 ***
x1            1.5363     0.3839   4.002 0.000125 ***
x2            0.7475     0.3852   1.940 0.055291 .
x3            3.1782     0.3813   8.336 5.87e-13 ***
x7            0.5479     0.3856   1.421 0.158652
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.074 on 95 degrees of freedom
Multiple R-squared:  0.5021,    Adjusted R-squared:  0.4811
F-statistic: 23.95 on 4 and 95 DF,  p-value: 1.026e-13

> ## calculate mse from true line
> mse.all <- sum((ans.all$fitted - true)^2)/n
> mse.sub <- sum((ans.sub$fitted - true)^2)/n
> mse.sw <- sum((ans.sw$fitted -true)^2)/n
> mse.sw1 <- sum((ans.sw1$fitted -true)^2)/n
>
```

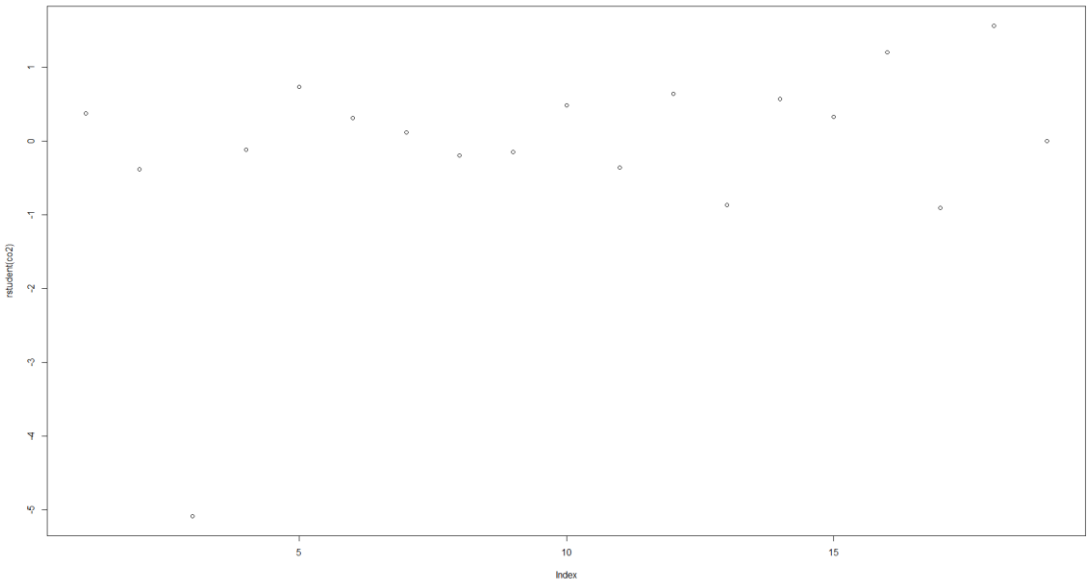
```
> mse.all
[1] 0.09172456
> mse.sub
[1] 0.03121932
> mse.sw
[1] 0.05449867
> mse.sw1
[1] 0.05449867
>

#################### Example 2 ######################
##### Variable Selection on coleman data #########
```

```
> coleman
  School  x1    x2     x3     x4    x5    y
1      1  3.83  28.87   7.20  26.60  6.19  37.01
2      2  2.89  20.10 -11.71  24.40  5.17  26.51
3      3  2.86  69.05  12.32  25.70  7.04  36.51
4      4  2.92  65.40  14.28  25.70  7.10  40.70
5      5  3.06  29.59   6.31  25.40  6.15  37.10

######################best subset selection#####################
X<-data.frame(cbind(coleman$x1,coleman$x2,coleman$x3,coleman$x4,coleman$x5))
X
y<-coleman$y
y
> ans.cp <- leaps(X, y, method="Cp", nbest=5)
> # Get the 5 "best" models according to Cp
> ans.cp$which[order(ans.cp$Cp)[1:5],]
     1      2      3      4      5
2  FALSE  FALSE   TRUE   TRUE  FALSE
3  TRUE   FALSE   TRUE   TRUE  FALSE
3  FALSE  FALSE   TRUE   TRUE  TRUE
4  TRUE   FALSE   TRUE   TRUE  TRUE
4  TRUE   TRUE    TRUE   TRUE  FALSE
> ans.cp$Cp[order(ans.cp$Cp)[1:5]]
[1] 2.832763  2.836093  4.613452  4.670224  4.797857
>
> ######################stepwise selection ####################
> upper <- formula(~ x1+x2+x3+x4+x5)
> lower <- formula(~ 1)
> #start with the "full" model.  (e.g. Backward Stagewise)
> ans0<-lm(y~1)
> ans1 <- lm(y ~ x1+x2+x3+x4+x5, data=coleman)
> ans.back <- step(ans1, direction = "backward")
Start:  AIC=34.05
y ~ x1 + x2 + x3 + x4 + x5

       Df Sum of Sq     RSS    AIC
- x2    1     2.884  63.122 32.987
- x5    1     3.433  63.671 33.160
```

```
<none>              60.238 34.051
- x1    1     9.096  69.334 34.864
- x4    1    28.184  88.422 39.728
- x3    1   153.800 214.038 57.408

Step:  AIC=32.99
y ~ x1 + x3 + x4 + x5

      Df Sum of Sq     RSS    AIC
- x5    1     0.714  63.835 31.211
<none>              63.122 32.987
- x1    1     8.361  71.483 33.474
- x4    1    25.495  88.616 37.772
- x3    1   191.544 254.665 58.884

Step:  AIC=31.21
y ~ x1 + x3 + x4

      Df Sum of Sq    RSS    AIC
<none>              63.84 31.211
- x1    1      8.59  72.43 31.737
- x4    1     26.13  89.96 36.073
- x3    1    507.00 570.83 73.027
>

> summary(ans.back)

Call:
lm(formula = y ~ x1 + x3 + x4, data = coleman)

Residuals:
   Min      1Q  Median      3Q     Max
-4.0812 -1.0017  0.1664  0.9797  4.3461

Coefficients:
             Estimate    Std. Error    t value      Pr(>|t|)
(Intercept)  12.11951    9.03643       1.341        0.199
x1           -1.73581    1.18290       -1.467       0.162
x3           0.55321     0.04907       11.273       5.06e-09 ***
x4           1.03582     0.40479       2.559        0.021 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.997 on 16 degrees of freedom
Multiple R-squared:  0.9007,    Adjusted R-squared:  0.8821
F-statistic: 48.38 on 3 and 16 DF,  p-value: 3.011e-08

#compare the models
> fit1<-lm(y~x1+x3+x4)
> fit2<-lm(y~x3+x4)
```

```
> fit3<-lm(y~x1+x3)
#compare the model with x1, x3, x4 and model with x3, x4
> anova(fit2, fit1)
Analysis of Variance Table

Model 1: y ~ x3 + x4
Model 2: y ~ x1 + x3 + x4
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     17 72.426
2     16 63.835  1    8.5911 2.1533 0.1616
> #p-value =0.1616>0.05, conclude  that Model 1 (x3 and x4) is adequate compared
#to Model 2.

> anova(fit1,fit3)
Analysis of Variance Table

Model 1: y ~ x1 + x3 + x4
Model 2: y ~ x1 + x3
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     16 63.835
2     17 89.960 -1   -26.125 6.5481 0.02102 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##model with x1 and x3 is not adequate.

########################delete #18 obeservation#####################
coleman2<-coleman[-18,]
######################best subset selection#####################
X2<-
data.frame(cbind(coleman2$x1,coleman2$x2,coleman2$x3,coleman2$x4,coleman2$x
5))
X2
y2<-coleman2$y
y2
ans.cp2 <- leaps(X2, y2, method="Cp", nbest=5)
# Get the 5 "best" models according to Cp

> ans.cp2$which[order(ans.cp2$Cp)[1:5],]
     1     2     3     4     5
5  TRUE  TRUE TRUE TRUE  TRUE
4 FALSE  TRUE TRUE TRUE  TRUE
4  TRUE FALSE TRUE TRUE  TRUE
3 FALSE FALSE TRUE TRUE  TRUE
3  TRUE FALSE TRUE TRUE FALSE
> ans.cp2$Cp[order(ans.cp2$Cp)[1:5]]
[1]  6.000000  8.145914  9.804895 11.367951 12.142464
>
######################stepwise selection ####################
upper <- formula(~ x1+x2+x3+x4+x5)
```

```
lower <- formula(~ 1)
#start with the "full" model.  (e.g. Backward Stagewise)
ans12 <- lm(y ~ x1+x2+x3+x4+x5, data=coleman2)
ans.back2 <- step(ans12, direction = "backward")
summary(ans.back2)
Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = coleman2)

Residuals:
   Min      1Q  Median      3Q     Max
-3.6972 -0.3152  0.1410  0.5365  1.6631

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.28738    9.31169   3.682  0.00276 **
x1          -1.61734    0.79431  -2.036  0.06264 .
x2           0.08544    0.03546   2.409  0.03153 *
x3           0.67393    0.06516  10.343 1.21e-07 ***
x4           1.10976    0.27902   3.977  0.00158 **
x5          -4.57076    1.43745  -3.180  0.00724 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.334 on 13 degrees of freedom
Multiple R-squared:  0.9633,    Adjusted R-squared:  0.9492
F-statistic: 68.27 on 5 and 13 DF,  p-value: 7.209e-09
> ########################outliers######################
> co2 <- lm(y ~ x1+x2+x3+x4+x5,data=coleman2)
> cop2=summary(co2)
> rstudent(co2)  ##gives rstudent values
         1          2          3          4          5          6
 0.37389758 -0.38020107 -5.08053003 -0.11488822  0.73346768  0.31551241
         7          8          9         10         11         12
 0.11717294 -0.19421781 -0.14839830  0.48763616 -0.35467246  0.64190777
        13         14         15         16         17         19
-0.86289406  0.56866741  0.33304775  1.20392506 -0.90440630  1.56168496
        20
 0.00475253
>outlierTest(co2)  ##Reports the Bonferroni p-value for the most extreme observation.
  rstudent        unadjusted p-value       Bonferonni p
3 -5.08053         0.00027041              0.0051379
```




#didn’t find x outliers and influential data point