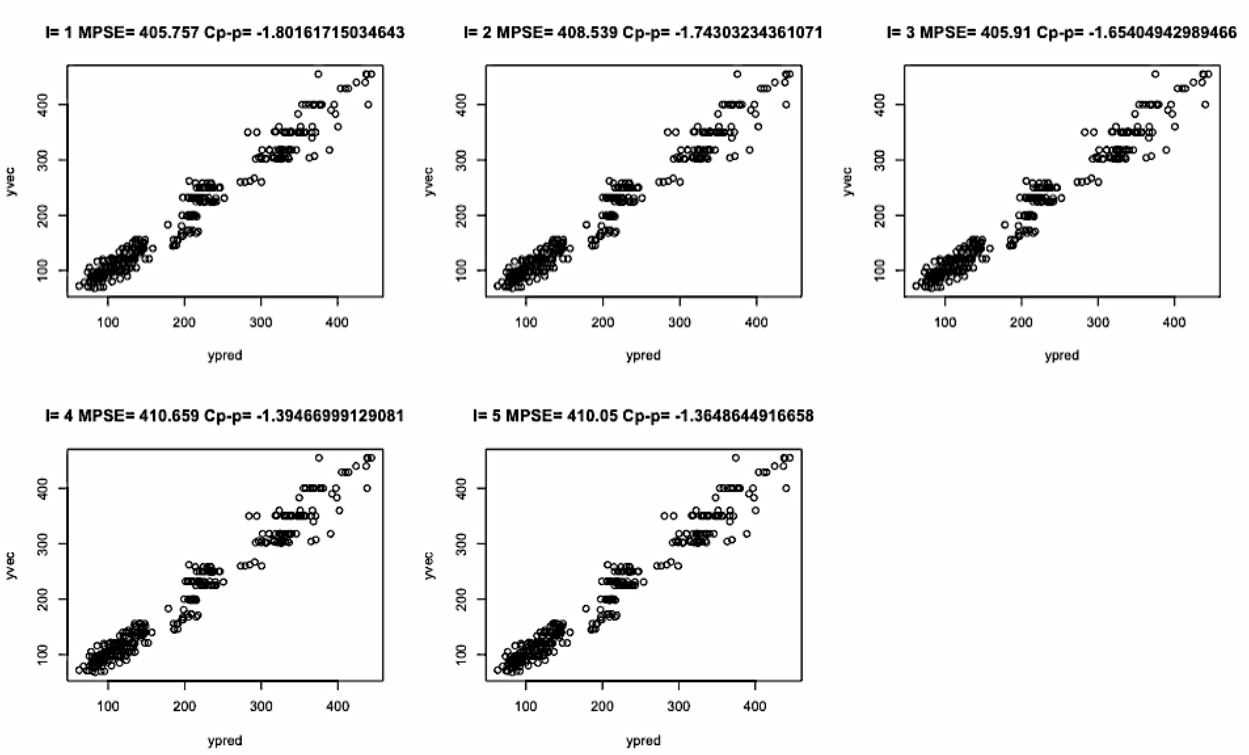
```
leaps.pck <-
 c("leaps.pck", "leaps.then.press.plot", "regpluspress", "matrix.2ndorder.make"
 )
leaps.then.press.plot <-
 function(xmat0a,yvec,ncheck=5,print.ls=F, resid.plot=F) #y regression, ncheck is the number of
models, ls is printing least square results from each of the ncheck, last prints residuals
 {
  xmat0<-as.matrix(xmat0a)
  n1<-ceiling(sqrt(ncheck)) #plot setup, calculate number of rows and columns --- no resid plots
  par(mfrow=c(n1,n1)) #set up a plotting region for all of the plots we print
  if(resid.plot){
   n1<-ceiling(sqrt(2*ncheck)) #plot setup, calc number of rows and columns --- with resid
plots
  }
  par(mfrow=c(n1,n1)) #set up plot region
  xmat<-matrix.2ndorder.make(xmat0) #create second order terms in matrix, only for pure
quads
  #xmat<-xmat0
  leaps.str<-leaps(xmat,yvec) #apply leaps alogrithm to xmat to find the regressions --- ouput
with incldue c_p
  z1<-leaps.str$Cp-leaps.str$size #extract c_p
  o1<-order(z1) #order the c_p values from smallest to largest
  matwhich<-(leaps.str$which[o1,])[1:ncheck,] #find the ncheck rows of whcih matrix with the
min c_p value
  z2<-z1[o1][1:ncheck]
  for(i in 1:ncheck){ #run a loop for ncheck values
   ls.str0<-regpluspress(xmat[,matwhich[i,]],yvec) #calculate press stat for the selection
variables
   if(print.ls){
    ls.print(ls.str0) #print out the regression results for this selection vairable
   }
   print(i)
   print(paste("Press=",ls.str0$press)) #print the press statistic
   parvec<-matwhich[i,] #take the ith row of variable selection matrix
   npar<-sum(parvec) #calculate how many regression matrix parameters
   print(paste("MPSE=",ls.str0$press/(length(yvec)-(npar+1)))) #print out on screen with all
decimals
   MPSE<-floor(1000*ls.str0$press/(length(yvec)-(npar+1)))/1000 #only print first 3 decimal
places
   print(paste("Cp-p=",z2[i])) #print c_p values
   ypred<-cbind(1,xmat[,matwhich[i,]])%*%ls.str0$coef #calculate predicted value

   plot(ypred,yvec,main=paste("I=",i,"MPSE=",MPSE,"Cp-p=",z2[i])) #plot true y value vs
predicted
```

```
    if(resid.plot){
      plot(ypred,ls.str0$resid) #plot residual vs. prediction
    }
   }

 }
regpluspress <-
 function(x,y){
  ls.str<-lsfit(x,y)
  reg.influence<-1/(1-hat(x))
  press<-sum((ls.str$resid/(1-hat(x)))^2)
  ls.str$leverage<-hat(x)
  ls.str$press<-press
  ls.str
 }
matrix.2ndorder.make <-
 function(x, only.quad=T){
  x0<-x
  dimn<-dimnames(x)[[2]] #extract the names of the variables
  num.col<-length(x[1,]) # how many columns
  for(i in 1:num.col){
   # if we are doing all 2nd order
   if(!only.quad){
    for(j in i:num.col){
     x0<-cbind(x0,x[,i]*x[,j])
     dimn<-c(dimn,paste(dimn[i],dimn[j],sep=""))
     #create interaction dimnames

    }
   }
   else{
    #in here only if doing only squared terms
    x0<-cbind(x0,x[,i]*x[,i])
    dimn<-c(dimn,paste(dimn[i],"2",sep="")) # squared dimmension names
   }
  }
  dimnames(x0)[[2]]<-dimn
  x0
 }
```


l= 1 MPSE= 405.757 Cp-p= -1.80161715034643
l= 2 MPSE= 408.539 Cp-p= -1.74303234361071
l= 3 MPSE= 405.91 Cp-p= -1.65404942989466
l= 4 MPSE= 410.659 Cp-p= -1.39466999129081
l= 5 MPSE= 410.05 Cp-p= -1.3648644916658
yvec
ypred
100
200
300
400

```
[1] 1
[1] "Press= 154593.709639015"
[1] "MPSE= 405.757768081403"
[1] "Cp-p= -1.80161715034643"
Residual Standard Error=19.2515
R-Square=0.9671
F-statistic (df=11, 380)=1015.677
p-value=0
```

	Estimate	Std.Err	t-value	Pr(>|t|)
Intercept	714.5343	481.7226	1.4833	0.1388
cylinders	26.3179	1.4506	18.1423	0.0000
horsepower	-1.5781	0.2639	-5.9791	0.0000
weight	0.0900	0.0136	6.6195	0.0000
acceleration	-17.7584	3.5619	-4.9857	0.0000
year	-13.3898	12.8017	-1.0459	0.2963
origin	-83.4081	11.5424	-7.2263	0.0000
horsepower2	0.0063	0.0009	7.3521	0.0000
weight2	0.0000	0.0000	-3.0021	0.0029
acceleration2	0.4141	0.1064	3.8906	0.0001
year2	0.0818	0.0841	0.9725	0.3314
origin2	18.4133	2.8856	6.3811	0.0000

```
[1] 2
[1] "Press= 155245.109052707"
[1] "MPSE= 408.53976066502"
[1] "Cp-p= -1.74303234361071"
Residual Standard Error=19.2539
R-Square=0.967
F-statistic (df=10, 381)=1116.86
p-value=0
```

	Estimate	Std.Err	t-value	Pr(>|t|)
Intercept	212.4116	39.9209	5.3208	0.0000
cylinders	26.2296	1.4484	18.1099	0.0000
horsepower	-1.5811	0.2640	-5.9903	0.0000
weight	0.0898	0.0136	6.6035	0.0000
acceleration	-18.2356	3.5330	-5.1616	0.0000
origin	-84.8880	11.4567	-7.4095	0.0000
horsepower2	0.0064	0.0009	7.4357	0.0000
weight2	0.0000	0.0000	-3.0294	0.0026
acceleration2	0.4285	0.1056	4.0585	0.0001
year2	-0.0062	0.0020	-3.0838	0.0022
origin2	18.7609	2.8668	6.5443	0.0000

```
[1] 3
[1] "Press= 154651.862739568"
[1] "MPSE= 405.910400891255"
[1] "Cp-p= -1.65404942989466"
Residual Standard Error=19.2603
R-Square=0.9672
F-statistic (df=12, 379)=930.2443
p-value=0
```

	Estimate	Std.Err	t-value	Pr(>|t|)
Intercept	826.9580	501.5901	1.6487	0.1000
cylinders	26.4471	1.4601	18.1136	0.0000
horsepower	-1.6251	0.2704	-6.0105	0.0000
weight	0.0855	0.0147	5.8198	0.0000
acceleration	-18.2356	3.6120	-5.0486	0.0000
year	-16.1187	13.2445	-1.2170	0.2244
origin	-82.4688	11.6059	-7.1058	0.0000
mpg2	-0.0047	0.0059	-0.8087	0.4192
horsepower2	0.0064	0.0009	7.3789	0.0000
weight2	0.0000	0.0000	-2.5384	0.0115
acceleration2	0.4275	0.1078	3.9670	0.0001
year2	0.1011	0.0875	1.1558	0.2485
origin2	18.2373	2.8951	6.2993	0.0000

```
[1] 4
[1] "Press= 155640.026560021"
[1] "MPSE= 410.65970068607"
[1] "Cp-p= -1.39466999129081"
Residual Standard Error=19.2612
R-Square=0.9671
F-statistic (df=11, 380)=1014.628
p-value=0
```

	Estimate	Std.Err	t-value	Pr(>|t|)
Intercept	239.5802	46.2694	5.1779	0.0000
cylinders	32.5093	8.4736	3.8365	0.0001
horsepower	-1.5339	0.2717	-5.6465	0.0000
weight	0.0849	0.0151	5.6043	0.0000
acceleration	-18.9013	3.6552	-5.1710	0.0000
year	-0.9167	0.3059	-2.9968	0.0029
origin	-84.5627	11.4606	-7.3786	0.0000
cylinders2	-0.5496	0.7310	-0.7518	0.4526
horsepower2	0.0062	0.0009	7.1382	0.0000
weight2	0.0000	0.0000	-2.3505	0.0193
acceleration2	0.4483	0.1092	4.1041	0.0000
origin2	18.6941	2.8673	6.5198	0.0000

```
[1] 5
[1] "Press= 155819.203141411"
[1] "MPSE= 410.05053458266"
[1] "Cp-p= -1.3648644916658"
```