```
R version 3.2.0 (2015-04-16) -- "Full of Ingredients"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> > options(STERM='iESS', str.dendrogram.last="'", editor='emacsclient', show.error.locations=TRUE)
>
> ff=function(x){ifelse(x<0,1,1+x)}
> xx0=seq(-2,2,len=1000)
> xx=(runif(10)-.5)*4
> yy=ff(xx)+rnorm(10,sd=.5)
> plot(xx,yy)
> fit4=lm(yy~xx+xx:I(xx>0))
> plot(xx,yy)
> points(xx0,predict(fit4,newdata=data.frame(xx=xx0)),pch=16,cex=.3,col='magenta')
> lines(xx0,ff(xx0),lty=2)
> source('./funs1.R')
> ls()
[1] "ff"       "findBest" "fit4"     "xx"       "xx0"      "yy"
> fit6=findBest(xx,yy)
> fit6
   -1.608     0.052     0.145     0.149     1.195     1.439
2.7272020 0.7819838 0.7975873 0.7996550 1.6222367 2.0463087
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                          (Intercept)
                                            0.8623057
                               I(x - 0.0520326560363173)
                                           -0.1443321
I(x - 0.0520326560363173):I(x > 0.0520326560363173)TRUE
                                            1.2753385

> xx=(runif(10)-.5)*4
> yy=ff(xx)+rnorm(10,sd=.5)
> plot(xx,yy)
> fit4=lm(yy~xx+xx:I(xx>0))
> points(xx0,predict(fit4,newdata=data.frame(xx=xx0)),pch=16,cex=.3,col='magenta')
> lines(xx0,ff(xx0),lty=2)
> fit6=findBest(xx,yy)
> fit6
   -0.161     0.572     0.886     1.038     1.229     1.392
0.7759441 1.4850454 1.6247950 1.7380456 1.8982850 2.0283370
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                          (Intercept)
                                            0.4605577
                               I(x - -0.161437639035285)
                                           -0.4486716
I(x - -0.161437639035285):I(x > -0.161437639035285)TRUE
                                            1.5097736

> fitB=attr(fit6,"bestFit")
> points(xx0,predict(fitB,newdata=data.frame(x=xx0)),pch=16,cex=.3,col='maroon')
> pdf(file='./hockeyFits2.pdf')
> dev.set(2)
quartz
     2
> legend(-1,2,legend=c("breakpt at 0","breakpt searched"),col=c("magenta","maroon"),lty=1,lwd=2)
> dev.set(3)
pdf
  3
> plot(xx,yy)
> points(xx0,predict(fit4,newdata=data.frame(xx=xx0)),pch=16,cex=.3,col='magenta')
> lines(xx0,ff(xx0),lty=2)
> points(xx0,predict(fitB,newdata=data.frame(x=xx0)),pch=16,cex=.3,col='maroon')
> legend(-1,2,legend=c("breakpt at 0","breakpt searched"),col=c("magenta","maroon"),lty=1,lwd=2)
> dev.off()
quartz
     2
> xx=(runif(10)-.5)*4
> yy=ff(xx)+rnorm(10,sd=.5)
> plot(xx,yy,pch=8)
> yy=ff(xx)+rnorm(10,sd=.75)
> plot(xx,yy,pch=8)
> resB=findBest(xx,yy)
> resB
   -1.152    -1.004    -0.408     0.513     0.571     1.242
 6.563164  5.968969  5.377043  7.811944  8.101674 10.752723
```

```
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                          (Intercept)
                                            0.1448622
                               I(x - -0.407797473482788)
                                           -1.2457184
I(x - -0.407797473482788):I(x > -0.407797473482788)TRUE
                                            2.6291471

> yy=ff(xx)+rnorm(10,sd=.75)
> resB=findBest(xx,yy)
> resB
   -1.152    -1.004    -0.408     0.513     0.571     1.242
3.2767309 2.5599479 0.6233902 3.4040831 3.5083899 5.0028865
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                          (Intercept)
                                           -0.3598191
                               I(x - -0.407797473482788)
                                           -1.5021071
I(x - -0.407797473482788):I(x > -0.407797473482788)TRUE
                                            3.0280017

> xx=(runif(10)-.5)*4
> yy=ff(xx)+rnorm(10,sd=.75)
> resB=findBest(xx,yy)
> resB
  -0.547   -0.366   -0.301   -0.207    0.495    1.002
9.112371 9.094348 9.065927 9.108533 8.682113 9.098324
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                        (Intercept)
                                          1.0297783
                             I(x - 0.494624975137413)
                                         -0.2797722
I(x - 0.494624975137413):I(x > 0.494624975137413)TRUE
                                          1.3091886

> plot(xx,yy,pch=8)
> yy=ff(xx)+rnorm(10,sd=.75)
> yy=ff(xx)+rnorm(10,sd=.5)
> plot(xx,yy,pch=8)
> findBest(xx,yy)
   -0.547    -0.366    -0.301    -0.207     0.495     1.002
1.7546476 1.7190971 1.6730111 1.5653681 1.7106260 0.8498422
attr(,"bestFit")

Call:
lm(formula = form, data = dd)

Coefficients:
                                      (Intercept)
                                        1.5455290
                           I(x - 1.00206525623798)
                                        0.2458207
I(x - 1.00206525623798):I(x > 1.00206525623798)TRUE
                                       10.0149394


> fit4=lm(yy~xx+xx:I(xx>0))
> fitB=attr(findBest(xx,yy),"bestFit")
> fitB

Call:
lm(formula = form, data = dd)

Coefficients:
                                      (Intercept)
                                           1.5455
                           I(x - 1.00206525623798)
                                           0.2458
I(x - 1.00206525623798):I(x > 1.00206525623798)TRUE
                                          10.0149

> lines(xx0,ff(xx0),lty=2)
> points(xx0,predict(fitB,newdata=data.frame(x=xx0)),pch=16,cex=.3,col='maroon')
> points(xx0,predict(fit4,newdata=data.frame(xx=xx0)),pch=16,cex=.3,col='green')
> lines(xx0,predict(fitB,newdata=data.frame(x=xx0)),pch=16,cex=.3,col='maroon')
> legend(-1,2,legend=c("breakpt at 0","breakpt searched"),col=c("green","maroon"),lty=1,lwd=2)
> pdf(file='./hockeyFits2.pdf')
> plot(xx,yy,pch=8)
> lines(xx0,ff(xx0),lty=2)
> legend(-1,3,legend=c("breakpt at 0","breakpt searched"),col=c("green","maroon"),lty=1,lwd=2)
> lines(xx0,predict(fitB,newdata=data.frame(x=xx0)),pch=16,cex=.3,col='maroon')
> points(xx0,predict(fit4,newdata=data.frame(xx=xx0)),pch=16,cex=.3,col='green')
> dev.off()
quartz
     2
```

```
> save(xx,yy,xx0,ff,file="hockeyStickV2.rda")
> install.packages('FNN')
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.fhcrc.org/bin/macosx/mavericks/contrib/3.2/FNN_1.1.tgz'
Content type 'application/x-gzip' length 102880 bytes (100 KB)
==================================================
downloaded 100 KB


The downloaded binary packages are in
        /var/folders/_q/jq0qydg52r7d5wk9360l1m2r0000gn/T//RtmplUbZe5/downloaded_packages
> library(FNN)
> resNN1=knn.reg(xx,test=xx0,y=yy,k=1)
Error in get.knnx(train, test, k, algorithm) :
  Number of columns must be same!.
> dim(xx)
NULL
> dim(as.matrix(xx))
[1] 10  1
> resNN1=knn.reg(as.matrix(xx),test=as.matrix(xx0),y=yy,k=1)
> length(resNN1$pred)
[1] 1000
> points(xx0,resNN1$pred,pch=16,cex=0.3,color="orange")
Warning message:
In plot.xy(xy.coords(x, y), type = type, ...) :
  "color" is not a graphical parameter
> points(xx0,resNN1$pred,pch=16,cex=0.3,col="orange")
> plot(xx,yy,pch=8)
Warning messages:
1: In getNamespace("grDevices") : "color" is not a graphical parameter
2: "color" is not a graphical parameter
> plot(xx,yy,pch=8)
> lines(xx0,resNN1$pred,pch=16,cex=0.3,col="orange")
>
2015-12-24 22:12:47.514 R[3781:6952087] The view <NSPrintThumbnailView: 0x7faaf443bd80> has a non-finite alignment frame of {{nan, 20}, {na

> resNN3=knn.reg(as.matrix(xx),test=as.matrix(xx0),y=yy,k=3)
> lines(xx0,resNN3$pred,pch=16,cex=0.3,col="firebrick")
>
```