Luke Beebe
Assignment 6 – General Smoothing

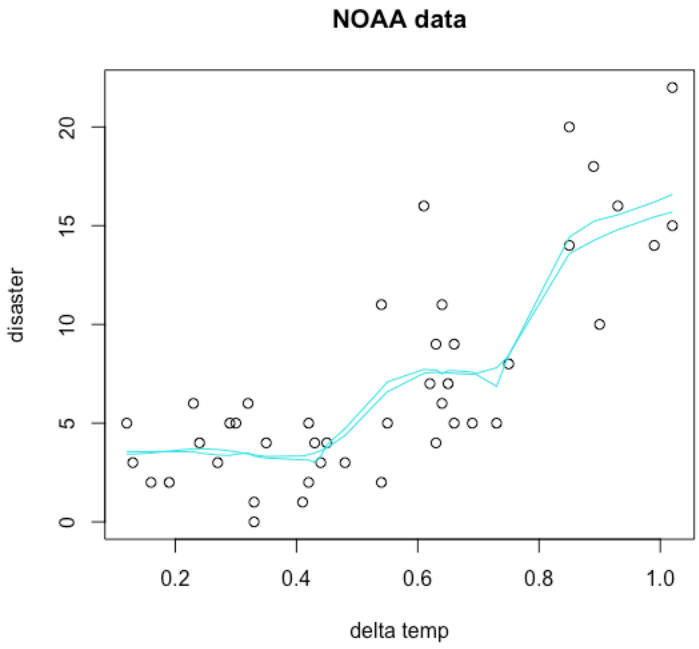
To start, I calculated the **PRESS** statistic in each smoother in smoother.pck and returned it from the function to use as reference to each model's accuracy given inputs.

```
PRESS<-sum((resid/(1-diag(smat)))^2)
```

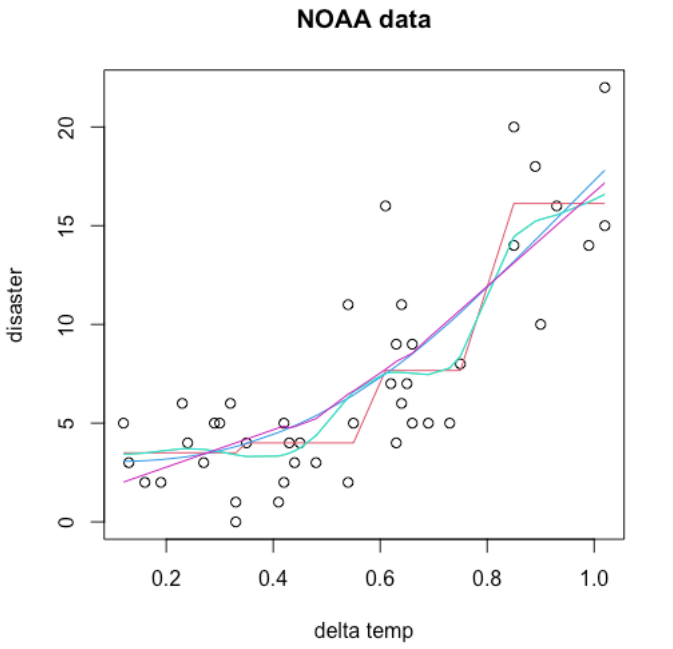
Next, I wrote a greedy random search, which takes the inputs **theta** and **nnn** for each model and adds/subtracts a random value to each input **num** amount of times, which resets if it finds inputs with a lower **PRESS** statistic. While tinkering through this code, I found I had to add a lot of bumpers so the smoother functions wouldn't crash while trying to do matrix multiplication. I got it pretty accurate as I tried multiple different inputs from theta=0.07 to theta=100, and it still found the best values. I believe this is because I made the standard deviation of the random number generator three times the value of the current most accurate theta value. This gave me the best results because if you start with a high theta, then the random number has a high variance. In essence, it starts with drastic changes and as theta shrinks, so does its spread.

```
greedy.random.searchA<-function(func,x,y,theta=0,nnn=0,nbin=0,num=10){ #inputs
  nnn0<-nnn
  nbin0<-nbin
  if(nnn<10){nnn<-10} #bumpers
  if(nnn>42){nnn<-42}
  if(nbin1<1){nbin1<-1} #bumpers
  if(nbin1>9){nbin1<-9}
  press0<-func(x,y,theta,nnn,nbin,do.plot=F)$press
  press00<-press0 #original press
  inc<-0
  theta0<-theta #original theta
  press1<-NA
  while(inc<num | press00==press0){
    epsilon<-rnorm(n=1,mean=0,sd=theta*3)
    theta1<-theta+epsilon
    nnn1<-nnn+ceiling(rnorm(n=1,mean=0,sd=5))
    nbin1<-nbin+ceiling(rnorm(n=1,mean=0,sd=3))
    if(nbin1<1 | is.na(nbin1)){nbin1<-1} #bumpers
    if(nbin1>9){nbin1<-9}
    if(nnn1<10 | is.na(nnn1)){nnn1<-10} #if nnn1 is NA, then set it to 10
    if(nnn1>42){nnn1<-42}
    if(theta1<0){theta1=theta1*(-1)} #if theta is negative, make it positive
    if(theta1<0.01){theta1=theta1+1} #if theta is too low, add 1
    press1<-func(x,y,theta1,nnn1,nbin1,do.plot=F)$press #new press statistic
    if(is.na(press1)){press1=press0} #if press statistic is NA, set it to previous best
    if(press1<press0){
      nbin<-nbin1
      press0<-press1
      inc<-0
      nnn<-nnn1
      theta<-theta1
    }else{
      inc<-inc+1
    }
  }
  func(x,y,theta,nnn,nbin,do.plot=T) #add best model to the graph
  if(nbin0>0){ #outputs nbin
    list(new.nbin=nbin,new.press=press0,old.nbin=nbin0,old.press=press00)
  }else if(nnn0>0){ #outputs values including nnn
    list(new.theta=theta,new.nnn=nnn,new.press=press0,old.theta=theta0,old.nnn=nnn0,old.press=press00)
  }else{ #outputs nbin
    list(new.theta=theta,new.press=press0,old.theta=theta0,old.press=press00)
```

One problem I ran into, mostly with the truncated smoother functions, was one of overfitting. Sometimes the most accurate models were obviously overfit producing sharp turns along the curves. An example is posted to the right. You can see the sharp turn left of 0.8 delta temp, where the model is compensating for that one point just below it.



Without the overfit models, I found the truncated and regular smoothers produce similar results, making me wonder if there is a benefit of the truncation. I consistently found the **theta** value as a better indicator of the accuracy of the model than **nnn**, except the last time I ran it where a more accurate model was found using **lambda** set to 4800, which was interesting. To the right is the comparison of the different smoothers and below is the code with approximations of the most accurate inputs for each smoother.




```
plot(NOAA.new$delta.temp,NOAA.new$X.disaster,xlab="delta temp",ylab="disaster",main="NOAA data")
bin.mean(NOAA.new$delta.temp,NOAA.new$X.disaster,nbin=4) #PRESS 377.3719
gauss.mean(NOAA.new$delta.temp,NOAA.new$X.disaster,lambda=0.07) #PRESS 442.7254
gauss.reg(NOAA.new$delta.temp,NOAA.new$X.disaster,lambda=0.2) #PRESS 469.9459
gauss.mean.trunc(NOAA.new$delta.temp,NOAA.new$X.disaster,lambda=0.07,nnn=34) #PRESS 442.7254
gauss.reg.trunc(NOAA.new$delta.temp,NOAA.new$X.disaster,lambda=4800,nnn=33) #PRESS 462.5478
```