Assessed Practical 3

Candidate number: 1035161

March 5, 2021

1 Data

The data set gives blood pressure measurements for some patients, taken at home and at the hospital by nurses. It consists of 22 observations and has 3 variables (*subject* identifies the patient, *home* and *hospital* give the values of the diastolic pressure (in mm Hg) for the subject, measured at home and at the hospital respectively).

It can be seen if Figure (1) that *home* has slightly lower median and bigger variance compared to *hospital*. A natural interpretation might be that blood pressure measurements at the hospital are taken by medical professionals and hence are more precise (i.e. have lower variance), and that patients are more relaxed at home and therefore have lower blood pressure (i.e. lower median/mean).

In Figure (2) we can see that *home* and *hospital* are strongly correlated (this was quite predictable as they are two measurements taken on the same subject), and that for most patients *hospital* is greater than *home* (most of the points are above the dotted line), visually supporting our intuition that at home blood pressure values tend to be lower.

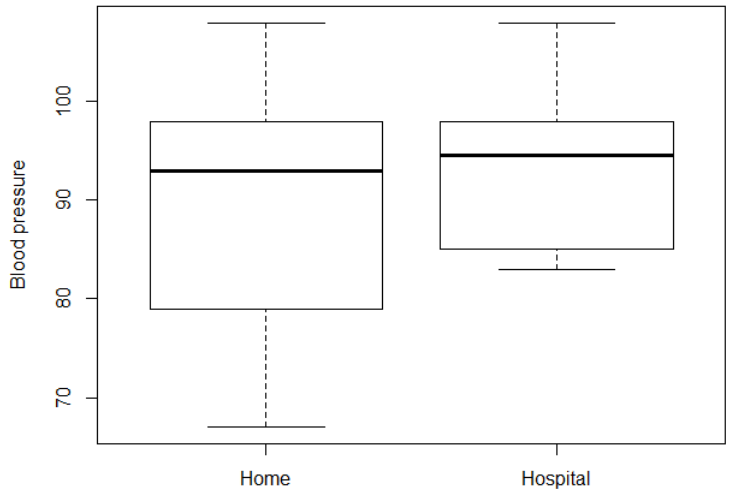

Figure (1): Box plot for *home* and *hospital* data

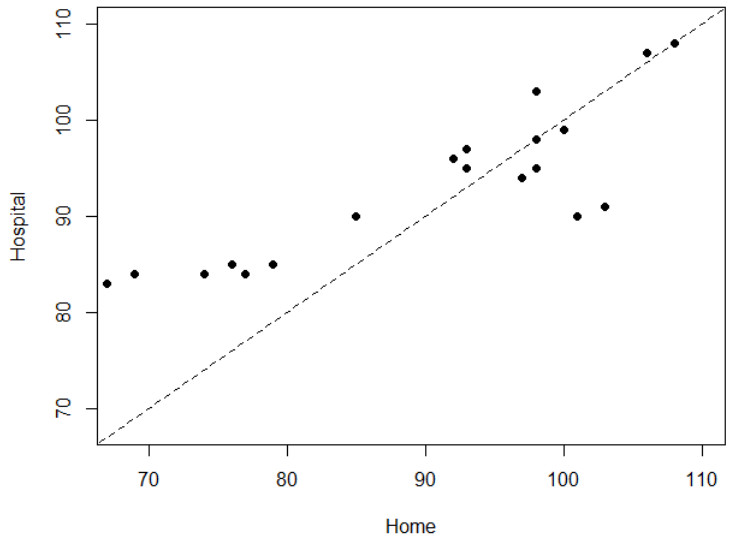

Figure (2): Scatterplot for *home* and *hospital* values for each observation

2 Smoothing

We would like to plot a sensible smooth for the data. We can start by thinking about Nadaraya-Watson Smoothers. There are very few data points and they are quite well-spaced so, as it can be seen in Figure (3), using a Box Kernel is not a good idea because the smooth will be very discrete (it looks like a step function) for whatever bandwidth we use.

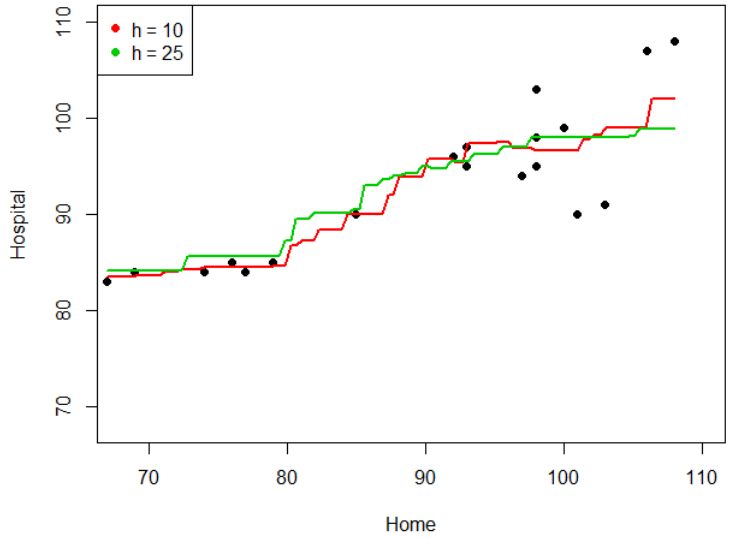

Figure (3): Smooths using Box Kernels of different bandwidths h for Nadaraya-Watson Smoothers

We can fix this issue by adopting a continuous Kernel (Normal Kernel in this case). By picking three arbitrary bandwidths we can already see in Figure (4) that N-W Smoothers with Normal Kernels give sensible results.

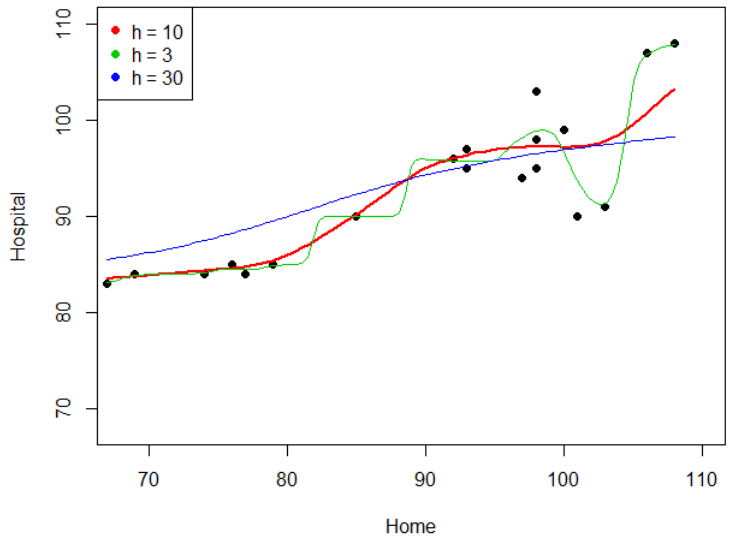

Figure (4): Smooths using Normal Kernel of different bandwidths h for Nadaraya-Watson smoothers

We can estimate the mean summed square error (MSSE) of a N-W Smoother with bandwidth h by using Leave-one-out Cross Validation (LOOCV). In Figure 5, it can be seen that according to this criterion the optimal bandwidth should be approximately 3. However, if we go back and look at the smooths in Figure (4), the green smooth (that has bandwidth 3) definitely looks like it is overfitting the data.

If we think carefully about the reason why this happens, the explanation is that the 4 right-most points are placed in a way that rewards excessively overfitting smoothers.

More precisely, we can see that about a third of the data points (those with low values of blood pressure, i.e. the left part of the plot) lie approximately on a straight line, therefore when we fit a model with one of these points removed in LOOCV, the squared error is not going to be significant. The biggest contribution to the MSSE in LOOCV is given when one of the 4 right-most points in the plot above is removed (let's call them *obs* 19, 20, 21, 22). Suppose we are calculat-

ing $m_h^{(-i)}$ where i is in (19, 20, 21, 22). When using a low bandwidth h, we "go through" the other three points almost perfectly. The values of *hospital* at these 4 points are (90, 91, 107, 108). Also $home_{20}$–$home_{19}$ = $home_{22}$–$home_{21}$ = 2. So the pairs {19, 20} and {21, 22} are very close together. If, say *obs* 19 is removed, then a low bandwidth smooth still passes very close to *obs* 20, this means that $(Y_{19} - m_h^{(-19)}(X_{19}))^2$ is small. The same logic applies to the pair {21, 22}. This is the reason why low bandwidth (the green line in Figure (4)) has optimal MSSE even though it is clearly overfitting the data.

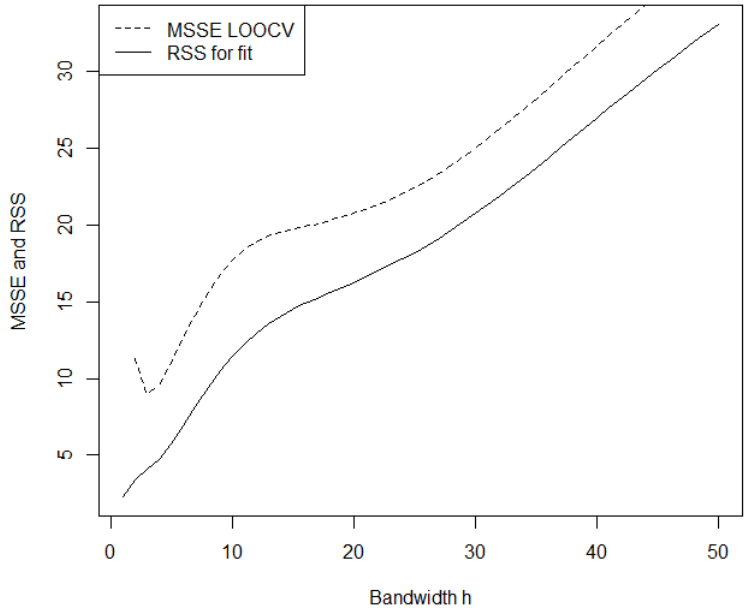

Figure (5): Estimated mean summed square errors (dotted) and residual sum of squares for a N-W smoothers with normal kernels of bandwidth $h \in \{1, 2, \ldots 50\}$

As a confirmation that our intuition (h = 10) is the most natural bandwidth for a N-W Smoother, if we fit a Local Linear Regression Smoother with bandwidth chosen according to the more sophisticated criterion in Ruppert, Sheather and Wand (1995) (using command *dpill*()), we can see that these two smooths are very similar.

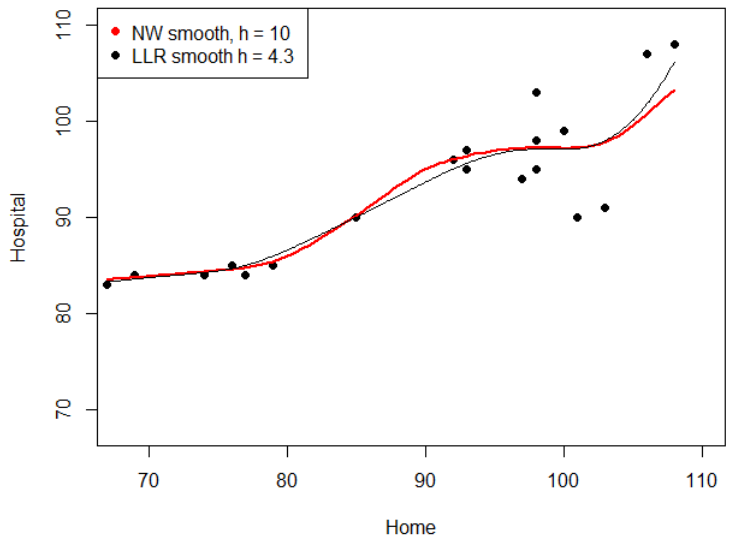

Figure (6): N-W smooth with bandwidth 10 (red), Local linear regression smooth with bandwidth 4.3 (black)

In addition, in this setting penalized regression is not a good idea because we have very few data points, hence even for a modest number of knots we would end up getting an overfitting spline passing through most of the data points.

We can conclude that a Nadaraya-Watson Smoother with Normal Kernel and bandwidth equal to 10 is a very sensible smoother for the data (Figure 6)

3 Hypotesis Testing

Let X_i, Y_i be the random variables describing the blood pressure of subject i measured at home and at the hospital respectively. It is reasonable to assume that X_i for $i \leq 22$ are iid and similarly that Y_i are iid. Clearly for a fixed i, X_i and Y_i are not independent (because they are two blood pressure measurements for the same individual). However, under the reasonable assumption of exchangeability between X and Y–Δ, we can easily test if Δ is zero. More formally:

Assumptions:

X_i *are iid* $\quad$ Y_i *are iid* $\quad$ (1)

$X_i,\ Y_i$–Δ $\quad$ *exchangeable for some* Δ $\quad$ (2)

$H_0 :\ \Delta = 0 \quad H_1 :\ \Delta \neq 0$

Let $R_i = rank(|\ X_i – Y_i\ |)$ *in* $\{|\ X_1 – Y_1\ |, ..., |\ X_n – Y_n\ |\}$ and our test statistic be

$W(I, R) = \sum_{i=1}^{22} R_i \cdot I_i \quad where\ I_i = Id(X_i \geq Y_i)$ $\quad$ (3)

Note that there are ties in the observed ranks r (we map all the tied values to the mean rank), hence under H_0 $R \nsim Unif(\mathcal{P}_{22})$.

However, under H_0 $I_i \sim Bernoulli(1/2)$ still holds because of exchangeability.

$(W(I, R) \mid H_0, R = r) \sim \sum_{i=1}^{22} r_i I_i \quad where\ I \sim U\{0, 1\}^n$

We can use Montecarlo simulation to approximate the following p-value (we simply need to simulate n independent Bernoulli(1/2) for every iteration). Hence for our 2-sided test:

$$p\ value = 2 \cdot min\{P(W(I, R) \leq w_{obs} \mid H_0, R = r), P(W(I, R) \geq w_{obs} | H_0, R = r)\}$$

$$= 2 \cdot min\{P(\sum_{i=1}^{22} r_i I_i) \leq w_{obs}), P(\sum_{i=1}^{22} r_i I_i) \leq w_{obs})\} \approx 0.023$$

Therefore we reject $H_0 :\ \Delta = 0$ as the p-value is below the standard 0.05 threshold.

Similarly, using again Montecarlo, we can find an approximate 95% confidence interval for the true delta value under assumptions (1), (2).

$\{\Delta_0 \; s.t. \; H_0 : \Delta = \Delta_0 \; is \; not \; rejected \; at \; a \; 0.05 \; level\} \approx (0.5, 5.9)$

4 Conclusions

After some exploratory data analysis, we tried to find a sensible smoother for *hospital* against *home*. Despite LOOCV, telling us to pick a very narrow bandwidth, by thinking about the properties of the data points, we chose a bigger and more natural bandwidth for the Nadaraya-Watson smoother. This decision was also supported by the fact that the local linear regression smooth with optimal bandwidth had a very similar behaviour. Finally, we rejected the hypothesis that blood pressure measurements taken at the hospital and measurements done at home had a zero offset, and gave a 95% confidence interval for the latter.

5 Appendix

```
df = read.table("http://www.stats.ox.ac.uk/~nicholls/CompStats
/bpdat.txt", header = TRUE)
df = df[(order(df$home)),]

boxplot(df$home, df$hospital, ylab = "Blood pressure",
names = c("Home", "Hospital"))
#we expect positive correlation
#we see hospital higher than home for most patients
plot(df$home, df$hospital, ylim = c(68,110), xlim = c(68,110),
 pch = 16, xlab = 'Home', ylab = 'Hospital')
lines(c(1:200),c(1:200),lty = 'dashed')
library(KernSmooth)

#trying box kernels
box_kernel1 <- ksmooth(df$home, df$hospital,kernel='box',
bandwidth=10)
box_kernel2 <- ksmooth(df$home, df$hospital,kernel='box'
,bandwidth=25)
plot(df$home, df$hospital, ylim = c(68,110), xlim =
c(68,110), pch = 16,
xlab = 'Home', ylab = 'Hospital')
lines(box_kernel1,col=2,lwd=2)
lines(box_kernel2,col=3,lwd=2)
```

```
legend("topleft",c("h = 10","h = 25"),col=c(2, 3), pch = 16)

#We can fix this issue by having a continuous kernel (normal kernel)
normal_kernel1 <- ksmooth(df$home, df$hospital,
kernel='normal',bandwidth=10)
normal_kernel2 <- ksmooth(df$home, df$hospital,
kernel='normal',bandwidth=3)#plug 55
normal_kernel3 <- ksmooth(df$home, df$hospital,
kernel='normal',bandwidth=30)
plot(df$home, df$hospital, ylim = c(68,110),
xlim = c(68,110), pch = 16,
xlab = 'Home', ylab = 'Hospital')
lines(normal_kernel1,col=2,lwd=2)
lines(normal_kernel2,col=3,lwd=1)
lines(normal_kernel3,col=4,lwd=1)
legend("topleft",c("h = 10","h = 3", "h = 30"),
col=c(2, 3,4), pch = 16)

#finding optimal bandwidth for LOOCV
h <- c(1:50)
mse <- NULL
for(i in 1:length(h)){
  k <- ksmooth(df$home, df$hospital,kernel='normal'
  ,bandwidth=h[i], x.points = df$home)
  if (i == 10){
    print(k$y - df$hospital)
  }
  if (i == 50){
    print(k$y - df$hospital)
  }

  mse[i] <- mean((k$y-df$hospital)^2)
}


mse_cv <- NULL
for(i in 1:length(h)){
  d<-rep(NA,length(df$home))
  #for each h knock out each point in turn and
  get (observed-predicted)^2
  for(j in 1:length(df$home)){
    k <- ksmooth(df$home[-j],df$hospital[-j],kernel='normal',
    bandwidth=h[i],x.points = df$home)
```

```
    d[j]<-(k$y[j]-df$hospital[j])^2
  }
  mse_cv[i] <- mean(d) #the LOO-CV estimate of the
   MSE for this h-value
}

#pdf("GN-LOOCV.pdf")
par(mar=c(4.5,4.5,1.5,1))
plot(h,mse,type='l', xlab='Bandwidth h',ylab='MSSE and RSS')

lines(h,mse_cv,lty=2); legend("topleft",c("MSSE LOOCV"
,"RSS for fit"),lty=c(2,1))


#optimal choice of bandwidth
(h.star<-h[which.min(mse_cv)])

plot(df$home, df$hospital, ylim = c(68,110), xlim = c(68,110),
 pch = 16, xlab = 'Home', ylab = 'Hospital')
lines(normal_kernel1,col=2,lwd=2)
#we see that we get a very similar smooth using optimal
#bandwidth criterion selection for a local regression, h = 4.3
k3 <- locpoly(df$home, df$hospital,bandwidth=dpill(df$home,
df$hospital),degree=1)
lines(k3,col=1,lwd=1.5)
legend("topleft",c("NW smooth, h = 10","LLR smooth h = 4.3 "),
col=c(2,1), pch = 16)



fnc = function(delta, df){
  df2 = data.frame(df)
  df2$diffs = df2$home - df2$hospital + delta
  df2$sign = rep(0,22)
  df2$sign[df2$diffs>0] = 1
  df2$diffs = abs(df2$home - df2$hospital + delta)
  df2$rank =  rank(df2$diffs)
  obs = sum(df2$sign*df2$rank)

  low = 0
  high = 0
  for(k in 1:10000){
    sim = numeric(22)
```

```
    for(j in 1:22){
      sim[j] = sample(c(0,1),1)
    }
    sim1 = sim*df2$rank
    sim2 = sum(sim1)

    if(sim2 <= obs){
      low = low +1
    }
    if(sim2 >= obs){
      high = high +1
    }
  }

  low = low/10000
  high = high/10000
  return(2*min(c(low,high)))
}

fnc(0.5,df)
fnc(6,df)
```