**EKT 720 Assignment 4**

1a) The objective of the nonparametric regression is to estimate the function that fits on the data directly without the need of estimating specified parameters which are estimated through the standard OLS estimation.

b) The underlying statistical model fits  
 y = f(x) + e (simple regression)  
with an unspecified f() function. Nonparametric regression aims to estimate that f() function is a continuous and smooth function. The error term, e, has an independent normal distribution with zero mean and variance is .

ci) Kernel estimation fits values by using locally weighted averaging, with the aid of some weight function where more weight is placed on data closer to the focal point, x0. The further away from the focal point, the less weight placed on that data point. The estimation is achieved by creating a neighbourhood around the focal point and weighting using that window.

cii) Local polynomial regression is similar to kernel estimation. The main difference is that it uses locally weighted regression by minimising the weighted sum of squares. It tends to be less bias that kernel estimation.

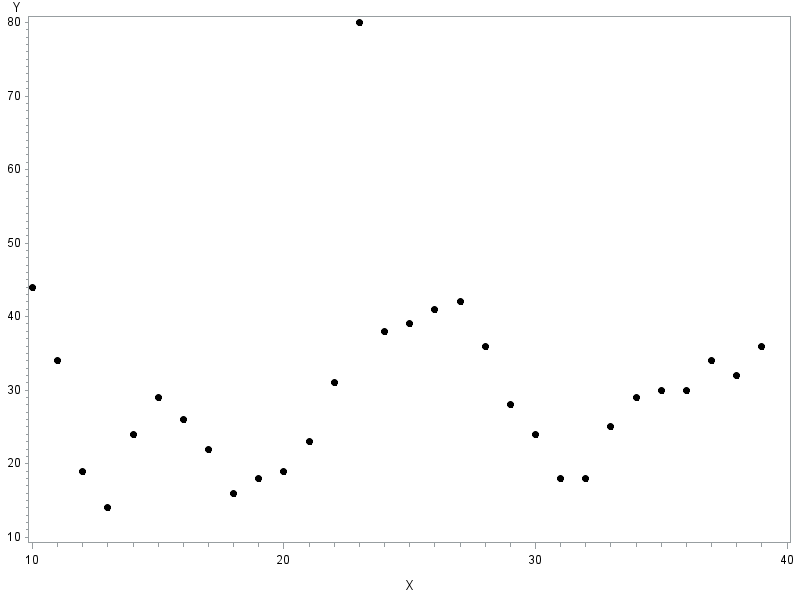
di) The tricube kernel calculates the weighted average with a bounded tail since it places a small weight to endpoints in the data window.

dii) The Gaussian kernel has a non-zero likelihood score and attempts to include all data points within the window, making it unbounded at the tails.

e)\*

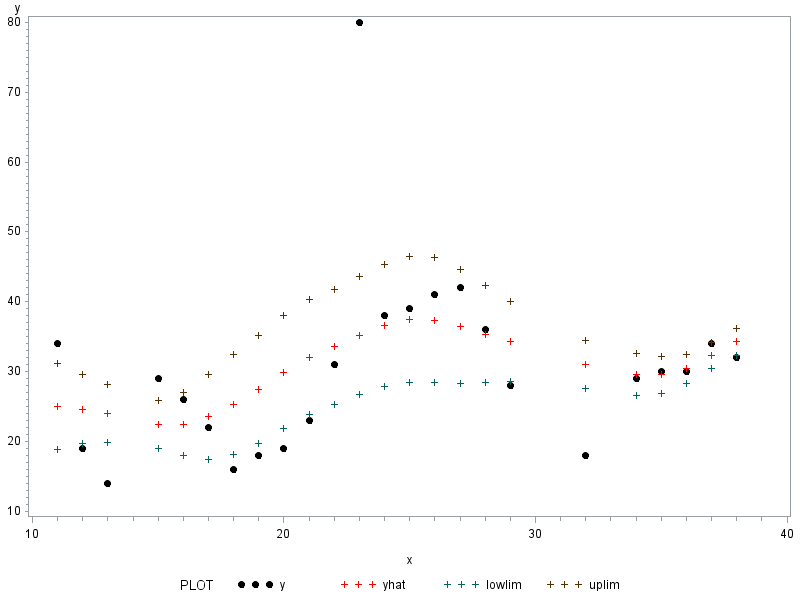
f) Nonparametric regression is useful in the estimation of non-linear data and helps in estimating the function directly onto the data instead of estimating parameters to estimate the trend.

2a)



2b) Since the given data is not of a linear nature, nonparametric regression is appropriate to aid in the estimation of the function so as to visualise the general trend without estimating the parameters.

d)



**Program:**

**data** a;

input X Y;

cards;

10 44

11 34

12 19

13 14

14 24

15 29

16 26

17 22

18 16

19 18

20 19

21 23

22 31

23 80

24 38

25 39

26 41

27 42

28 36

29 28

30 24

31 18

32 18

33 25

34 29

35 30

36 30

37 34

38 32

39 36

;

**run**;

**proc** **gplot** data=a;

plot y\*x;

**run**;

**proc** **iml**;

use a;

read all into xy;

n=nrow(xy);

span=**0.7**;

n1=**0.8**\*n;

m=round(span\*n1); /\*number of poins to include in the window\*/

randsam=j(n,**1**,**1**);

randgen=rannor(randsam)||xy;

call sort (randgen, {**1**});

xy1=randgen[**1**:n1,**2**:**3**];

rows\_xy1=nrow(xy1);

\*nxy=nrow(xy1);

\*print xy1; /\*checkpoint\*/

minxy1=min(xy1[,**1**]);

maxxy1=max(xy1[,**1**]); /\*can also use sort instead\*/

\*\*print minxy1 maxxy1;

/\*kernel\*/

do x\_focal=**1** to rows\_xy1;

xc=xy1[,**1**]-xy1[x\_focal,**1**];

xd=abs(xc);

mhood=xy1||xc||xd; /\*neighborhood\*/

call sort(mhood,{**4**}); /\*sort according to abs differences\*/

mhood1=mhood[**1**:m,];

h=**0.5**\*(max(mhood1[,**1**])-min(mhood1[,**1**])); /\*half the length of the window\*/

z=mhood1[,**4**]/h; /\*equation for abs(z) in the tricube weight\*/

wt=((**1**-(z)##**3**)##**3**)#(z<**1**); /\*condition matrix filled with 1's and 0's\*//\*'##'=element power\*/

\*\*print wt; /\*checkpoint\*/

y\_hat=(wt`\*mhood1[,**2**])/(j(**1**,m,**1**)\*wt); /\*calculation of weighted averages of the y values\*/

\*\*ker =ker//(x\_focal||y0\_hat);

\*\*print y0\_hat;

/\*\*LPR with df=m-2\*\*/

w=diag(wt);

X=J(nrow(mhood1),**1**,**1**)||mhood1[,**3**];

Y=mhood1[,**2**];

bhat\_lpr=inv(x`\*w\*x)\*x`\*w\*y;

bhat=bhat//bhat\_lpr[**1**];

yhat=x\*bhat\_lpr;

\*\*print yhat;

sse=(y-yhat)`\*w\*(y-yhat);

mse=sse/(m-**2**);

varb=mse\*inv(x`\*w\*x);

k=tinv(**0.975**, m-**2**);

lowlim=lowlim//(bhat\_lpr[**1**]-(k\*(sqrt(varb[**1**,**1**]))));

uplim=uplim//(bhat\_lpr[**1**]+(k\*(sqrt(varb[**1**,**1**]))));

end;

lprmtrx=xy1||bhat||lowlim||uplim;

print lowlim uplim;

create assnglpr4 from lprmtrx[colname={'x' 'y' 'yhat' 'lowlim' 'uplim'}];

append from lprmtrx;

/\*create assngkern4 from ker[colname={'x\_focal' 'y\_hat'}];\*/

**quit**;

goptions reset = all;

symbol1 v=dot c=black h=**1**;

symbol2 c=red h=**1**;

/\*proc gplot data=assngkern4;

plot y\_hat\*x\_focal/overlay legend;

run;\*/

**proc** **gplot** data=assnglpr4;

plot (y yhat lowlim uplim)\*x/overlay legend /\*vaxis=2 haxis=2\*/;

**run**;